



**MODELING, STABILITY, AND CONTROL OF A ROTATABLE TAIL ON A  
MICRO AIR VEHICLE**

THESIS

Travis J. Higgs, Captain, USAF  
AFIT/GAE/ENY/06-05

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

**AIR FORCE INSTITUTE OF TECHNOLOGY**

**Wright-Patterson Air Force Base, Ohio**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GAE/ENY/06-05

**MODELING, STABILITY, AND CONTROL OF A ROTATABLE TAIL ON A  
MICRO AIR VEHICLE**

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Aeronautical Engineering

Travis J. Higgs, BS

Captain, USAF

December 2005

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GAE/ENY/06-05

**MODELING, STABILITY, AND CONTROL OF A ROTATABLE TAIL ON A  
MICRO AIR VEHICLE**

Travis J. Higgs, BS

Captain, USAF

Approved:

\_\_\_\_\_  
Dr. Mark F. Reeder (Chairman)

\_\_\_\_\_  
Date

\_\_\_\_\_  
Paul A. Blue, Major, USAF (Member)

\_\_\_\_\_  
Date

\_\_\_\_\_  
Dr. Richard G. Cobb (Member)

\_\_\_\_\_  
Date

## **Abstract**

This research uses existing experimental wind tunnel data to develop a non-linear model that is used to characterize the stability of a flexible wing Micro Air Vehicle (MAV) with a rotatable tail. The experimental data are curve fit based on either angle of attack or angle of sideslip, and the coupled effect of tail rotation and tail deflection on the force and moment coefficients. Static optimization trims the input and state variables for Steady Level Unaccelerated Flight (SLUF). The resulting initial conditions are fed to an open loop non-linear Simulink/Matlab simulation. The study found that the bare MAV design is unstable, but parametric studies identified practical modifications that could be made to the MAV to improve its open loop stability characteristics. The study found that the coupling affect due to the dihedral derivative,  $Cl_\beta$ , played a large role in destabilizing the lateral-directional dynamics and a feedback Stability Augmentation System is required for flight.

## **Acknowledgments**

I would like to thank my faculty advisor, Dr. Mark Reeder for his motivation and support throughout this study. I also want to express my true appreciation to Major Paul Blue for his guidance, dedication, and availability throughout the pursuit of this challenge. I owe my deepest gratitude to my family and friends who have supported and encouraged me throughout this research program.

Travis J. Higgs

## Table of Contents

	Page
Abstract.....	iv
Acknowledgments.....	v
Table of Contents.....	vi
List of Figures .....	ix
List of Tables .....	xi
List of Symbols and Abbreviations .....	xii
 I. Introduction .....	 1
Background.....	1
Problem Statement.....	5
Research Objectives.....	5
Assumptions/Limitations .....	7
Preview .....	8
 II. Literature Review .....	 9
Chapter Overview .....	9
Relevant Research.....	9
Multiple Regression.....	10
Forces and Moments .....	11
Optimization .....	16
Nonlinear Equations of Motion .....	17

	Page
Nonlinear Solver .....	19
Linearization and Open Loop Stability .....	20
III. Methodology .....	23
Chapter Overview .....	23
Curve Fits.....	24
Forces and Moments .....	28
Trimmer .....	29
EOM Solver .....	32
Linearization .....	34
IV. Results and Analysis.....	35
Chapter Overview .....	35
Curve Fits.....	35
Open Loop Simulation.....	49
Artificial Stability Improvements .....	54
V. Conclusions and Recommendations .....	71
Chapter Overview .....	71
Conclusions of Research.....	71
Recommendations for Future Research .....	72
Roll Control. ....	73
MIMO Feedback Control.....	73
Alternate Tail Configuration.....	74



	Page
Appendix A: Experimental Data for Tail Effectiveness .....	76
Appendix B: Tail Effectiveness Curve Fit Matlab Code .....	77
Appendix C: Matlab Force and Moment Calculator.....	96
Appendix D: Matlab Trimmer for SLUF.....	98
Appendix E: Nonlinear EOMs.....	103
Appendix F: Dynamic Output (Flight Path & Strip Charts) .....	106
Appendix G: Matlab Linearization and Eigenvalues.....	111
Appendix H: Formatted State Space System .....	113
Appendix I: Modified Data for $C_l$ , $C_n$ , and $C_Y$ (Zeroed and Mirrored).....	114
Bibliography .....	115

## List of Figures

	Page
Figure 1. Rigid and Flexible Wing BATCAM V-Tail Variants .....	2
Figure 2a. Spanwise Storage Savings Using a Flexible Wing.....	3
Figure 2b. Latest Configuration of the MAV .....	3
Figure 3. Latest Configuration of the MAV .....	7
Figure 4. Physical Explanation for Rolling Moment due to Roll Rate .....	13
Figure 5. Tail Deflection and Rotation Sign Convention .....	24
Figure 6. Dynamic Open Loop Simulink Model .....	32
Figure 7. $C_{D\alpha}$ For Zero Tail Inputs.....	36
Figure 8. $C_{L\alpha}$ For Zero Tail Inputs .....	37
Figure 9. $C_{m\alpha}$ For Zero Tail Inputs.....	37
Figure 10. $C_{l\beta}$ For Zero Tail Inputs .....	38
Figure 11. $C_{n\beta}$ For Zero Tail Inputs.....	39
Figure 12. $C_{Y\beta}$ For Zero Tail Inputs .....	40
Figure 13. Tail Mounting Asymmetry .....	41
Figure 14. Tail Effect on Drag Coefficient.....	43
Figure 15. Tail Effect on Lift Coefficient.....	44
Figure 16. Tail Effect on Pitching Moment Coefficient .....	45
Figure 17. Tail Effect on Roll Moment Coefficient .....	46
Figure 18. Tail Effect on Yaw Moment Coefficient.....	47
Figure 19. Tail Effect on Side Force Coefficient.....	48

	Page
Figure 20. True Airspeed Effect on Trim Angles .....	50
Figure 21. Flight Path Results of Unbiased Data Unaugmented MAV .....	51
Figure 22. Strip Chart Results of Unbiased Data Unaugmented MAV .....	52
Figure 23. Crucial Strip Results of Unbiased Data Unaugmented MAV .....	53
Figure 24. Linearized Eigenvalue Stability Improvement as a Function of $C_{n\beta}$ and $C_{lp}$ . .....	58
Figure 25. Flight Path for DF=50 and $C_{n\beta} = +0.0018$ .....	59
Figure 26. Strip Charts for DF=50 and $C_{n\beta} = +0.0018$ .....	60
Figure 27. Crucial States for DF=50 and $C_{n\beta} = +0.0018$ .....	61
Figure 28. Modified $C_l$ Curve Fit for $\delta_m=0^\circ$ Bias Removal & Mirroring .....	63
Figure 29. Modified $C_n$ Curve Fit for $\delta_m=0^\circ$ Bias Removal.....	64
Figure 30. Modified $C_Y$ Curve Fit for $\delta_m=0^\circ$ Bias Removal .....	65
Figure 31. Flight Path for $\delta_m=0^\circ$ Bias Removal & Mirroring .....	66
Figure 32. 3D Flight Path for $\delta_m=0^\circ$ Bias Removal & Mirroring .....	67
Figure 33. Strip Charts for $\delta_m=0^\circ$ Bias Removal & Mirroring.....	68
Figure 34. Crucial States for $\delta_m=0^\circ$ Bias Removal & Mirroring.....	69

## List of Tables

	Page
Table 1. Aircraft Physical Properties .....	5
Table 2. Choices for State Variables.....	17
Table 3. A Matrix & Eigenvalues of Unbiased Data Unaugmented MAV .....	54
Table 4. Parametric Lateral-Directional Sensitivity Analysis .....	56
Table 5. A Matrix and Eigenvalues for $\delta_m=0^\circ$ Bias Removal & Mirroring .....	70

## List of Symbols and Abbreviations

$\eta$	East Position
$\gamma$	Flight Path Angle
$\xi$	North Position
A/C	Aircraft
AR	Aspect Ratio
$b$	Wing Span
BATCAM	combat CAMera
$\bar{c}$	Wing Mean Chord
CAS	Control Augmentation System
$C_D$	Drag Coefficient
$cg$	Center of Gravity
$C_L$	Lift Coefficient
$C_l$	Roll Moment Coefficient
$Cl_\beta$	Roll Damping Due to Roll Rate
$C_m$	Pitch Moment Coefficient
$C_n$	Yaw Moment Coefficient
$C_Y$	Side Force Coefficient
$D$	Drag Force
deg	Degrees
GWOT	Global War On Terrorism
$h$	Altitude
$I_{xx}$	Roll Mass Moment of Inertia
$I_{xz}$	Product of Inertia
$I_{yy}$	Pitch Mass Moment of Inertia
$I_{zz}$	Yaw Mass Moment of Inertia
$L$	Lift Force
$l$	Roll Moment
$M$	Mach Number
$m$	Pitch Moment
MAV	Micro-Air-Vehicle
MIMO	Multiple Input/Multiple Output
mph	Miles Per Hour
$n$	Yaw Moment
$p$	Roll Rate (Flight Path Components)
$q$	Pitch Rate (Flight Path Components)
$r$	Yaw Rate (Flight Path Components)
$R^2$	Correlation Coefficient
rad	Radians
$S$	Wing Area
SAS	Stability Augmentation System
SLUF	Steady Level Unaccelerated Flight
$U_\infty$	Free Stream Velocity

$V_T$	True Airspeed
$Y$	Side Force
$\beta$	Sideslip Angle
$\delta_a$	Aileron Deflection Angle
$\delta_e$	Tail Deflection Angle
$\delta_r$	Rudder Deflection Angle
$\delta_{rn}$	Tail Rotation Angle
$\mu$	Bank Angle about velocity vector
$\rho$	Air Density
$\chi$	Heading Angle

# MODELING, STABILITY, AND CONTROL OF A ROTATABLE TAIL ON A MICRO AIR VEHICLE

## I. Introduction

### Background

Timely and accurate reconnaissance is crucial when dealing with the fog and friction of war. Today's urban and asymmetrical combat environment emphasizes the need for more sensors throughout the battlefield to help identify targets and provide feedback following a strike. Large Unmanned Aerial Vehicles (UAVs) such as the *Global Hawk* or *Predator* are not well suited for immediate response to reconnaissance needs at the small unit level. A newer class of Micro Air Vehicles (MAVs) can help fill the void in the information mesh on the tactical level. These compact, man-portable systems can be carried, deployed, and operated by a single troop to provide valuable real-time information that could be critical to mission success by determining what is over the hill or down the street.

The Air Force Research Lab, Munitions Directorate, Flight Vehicles Integration Branch (AFRL/MNAV) developed a man-portable, carbon-fiber matrix MAV designated ComBAT-CAMera (BATCAM) for field deployment in the Global War on Terrorism (GWOT). An early variant used a rigid carbon-fiber wing with a V-tail configuration. Captain Anthony Deluca (USAF) performed wind tunnel analysis to compare the next variant's flexible wing performance with that of the original rigid design. The two wing

variants are displayed in Figure 1. The rigid wing model is in the lower left and the flexible wing model is in the upper right. The investigation found that "...the flexible-wing vehicle had an improved lift-to-drag ratio over a wide range of angles of attack. The lift data also suggested that the wing flexure would lead to a steadier flight when the vehicle is subjected to variation in the headwind due to gusts" (Deluca, 2004:97).



Figure 1. Rigid and Flexible Wing BATCAM V-Tail Variants  
(Photo courtesy of Dr. Richar Cobb, AFIT/ENY)

The flexible wing's primary advantage is the ability to curl down around the fuselage resulting in more than 60 percent reduction in spanwise storage requirements. Figure 2a illustrates the space savings of the flexible wing configuration.



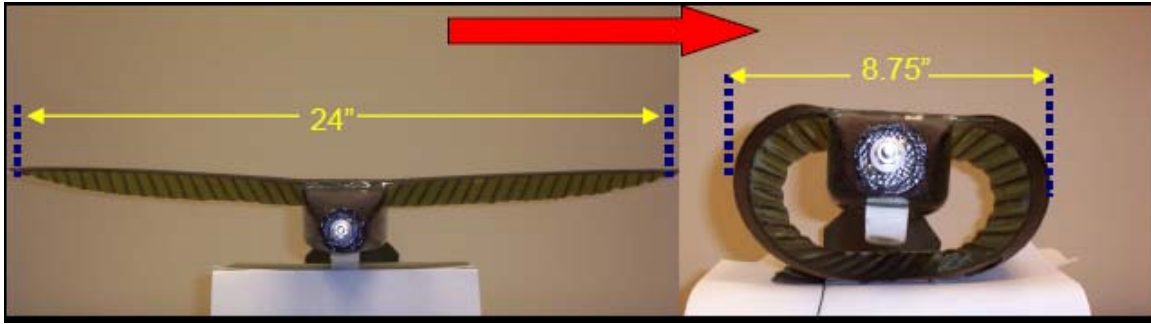


Figure 2a. Spanwise Storage Savings Using a Flexible Wing (Parga, 2004:7)

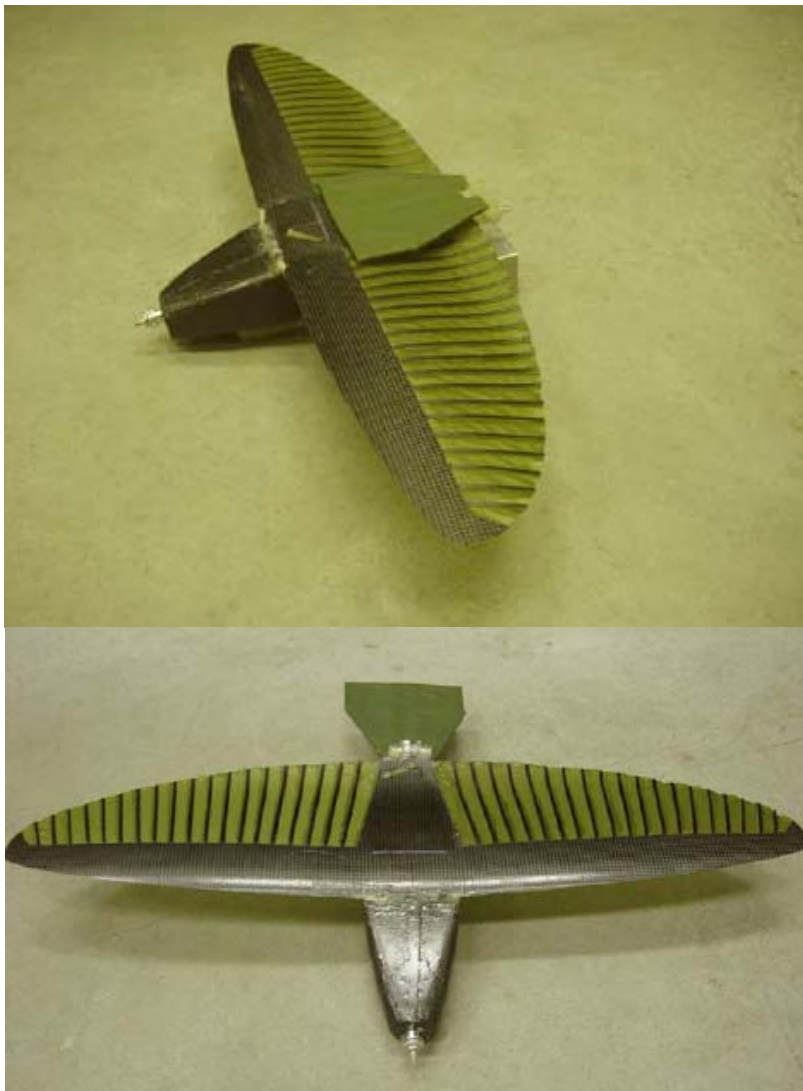


Figure 2b. Latest Configuration of the MAV (Leveron, 2005:5,16)

The spanwise storage reduction led to considerations for reducing the stowed length of the aircraft. The concept of a rotatable tail showed promise for more than a 50 percent reduction in storage length while maintaining the possibility for controlled flight via bird-like tail movements. Lieutenant José Rivera Parga (Mexican Navy) executed a wind tunnel investigation of the static stability and control effectiveness of the flexible wing rotatable tail MAV configuration. The "... data suggested that a rotatable tail could provide enough control authority to turn the vehicle, but that additional vertical tail surface would be required on the modified vehicle to maintain directional stability" (Parga, 2004:190).

Ensign Troy Leveron (US Navy) performed further wind tunnel investigation of tail position effect on force and moment coefficients over a broader range of settings than the previous works. This research confirmed a lack of yaw stability due to the absence of a vertical stabilizer, and led to a modified configuration that consisted of "two ventral fins placed on the lower aft portion of the MAV fuselage" (Leveron, 2005:5). These fins successfully provided a small degree of static yaw stability as expected. Figure 2b is a pictorial description of the latest iteration of the MAV with the tail folded forward in the stowed position in the upper picture and fully extended on the bottom. This variant has extended dimensions as follows: flexible wing span of 24.0 inches, a root chord of 6.0 inches, and length of 18.2 inches. Additional MAV physical properties of interest for this study are listed in Table 1.

Table 1. Aircraft Physical Properties.

Property Name	Matlab Variable	Value with Units
Weight	W	0.79 lb <sub>f</sub>
Mass	M	0.025 slugs
Roll Mass Moment of Inertia	$I_{xx}$	0.001934 slug ft <sup>2</sup>
Pitch Mass Moment of Inertia	$I_{yy}$	0.006631 slug ft <sup>2</sup>
Yaw Mass Moment of Inertia	$I_{zz}$	0.007020 slug ft <sup>2</sup>
Product of Inertia	$I_{xz}$	0
Planform Area	S	0.65 ft <sup>2</sup>
Wing Span	B	2.0 ft
Wing Mean Geometric Chord	Cbar	0.35 ft

The data collected in the previous studies of this aircraft were used primarily for static stability analysis and control effectiveness. The next step in rotatable tail development was to use the existing wind tunnel data to create mathematical models to predict the dynamic behavior of the MAV.

### Problem Statement

This research uses existing experimental wind tunnel data to develop a non-linear aircraft model that can be used analyze the dynamic motion of a flexible wing MAV with a rotatable tail.

### Research Objectives

This research project began with the following objectives:

- A. Develop mathematical curve fits to existing experimental force and moment coefficient data
- B. Build up non-linear aircraft (A/C) force and moment equations and incorporate the curve fits.

- C. Develop a 6 Degree of Freedom (6DOF) model using the forces and moments and the non-linear A/C EOMs
- D. Develop optimization code to determine the necessary control inputs and initial states for trimmed Steady, Level, Unaccelerated Flight (SLUF)
- E. Verify simulation model by performing SLUF, glide, steady level turns, and other basic maneuvers
- F. Use dynamic optimization to define the vehicle's flight envelope

The only objectives not met were the glide and steady level turn portions of the simulation verification in objective E and the dynamic optimization of objective F. A flow chart of overall research process of pursuing these objectives is shown in Figure 3. Note that when SLUF was not successfully achieved, the flow steps back through the process to improve and verify the building blocks toward simulated flight.

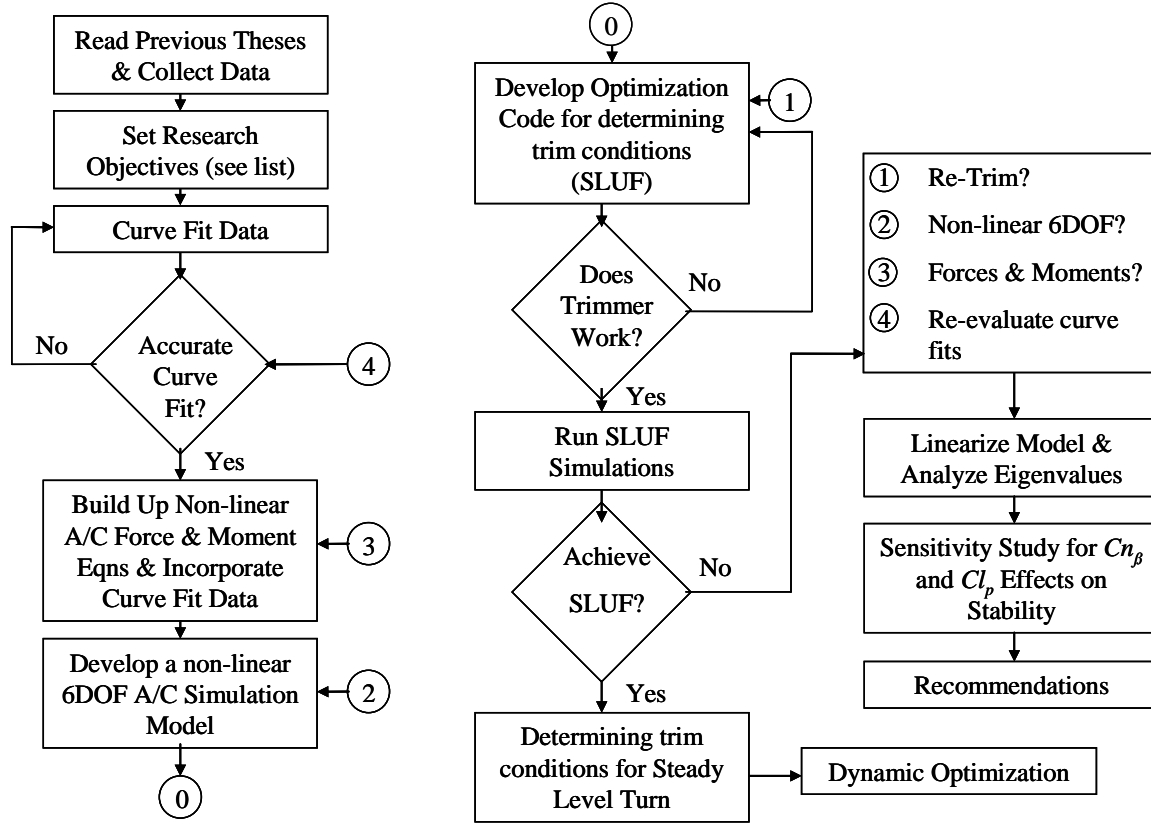


Figure 3. Research Flow Chart

### Assumptions/Limitations

The following assumptions are used to simplify and bound pursuit of the research objectives:

- A. Thrust is not characterized as a function of throttle setting. It is only considered an axial force along the x-axis of the aircraft. There is no accounting for moments or downwash effects associated with a spinning propeller.
- B. The product of inertia,  $I_{xz}$ , is an aircraft property that describes the asymmetric inertial characteristics of an aircraft, and is assumed zero

throughout simulations conducted in this research. Large tail deflections and rotations may create small inertial asymmetries during actual flight that should improve aircraft response in the desired direction of the control input.

- C. The aircraft center of gravity ( $cg$ ) is assumed constant and equal to that of the data collected during the wind tunnel tests. This assumption should remain valid due to no fuel burn or expendables associated with the battery powered MAV.
- D. Other factors purposely disregarded in this study are actuator rate saturation and aerobatic maneuvers.

## **Preview**

The motivation, historical development, and limitations of this study have been detailed. Chapter II sets forth the background theory and pertinent research to this topic. Chapter III describes the method in which the research objectives were pursued. Chapter IV presents the results and analysis of the various components of this investigation. Finally, Chapter V makes conclusions about the predicted stability of the MAV and suggests future areas of development and study.

## II. Background

### Chapter Overview

This chapter sets forth a limited review of bird tail control within the aeronautics field and lays the theoretical foundation of the methods used to achieve the research goals. Topics within this chapter include statistical explanation of the least squares method, explanation of the forces and moments about an aircraft, optimization theory, simultaneous non-linear equations and their solutions, and control theory for MAVs.

### Relevant Research

Throughout history, birds have been the inspiration behind man's desire for flight. Though controlled powered flight has been reality for more than 100 years, bird-like flight controls remain an elusive mystery in the field of aeronautics. In 1992, Robert G. Hoey published *Research on the Stability and Control of Soaring Birds* in which he successfully built and flew a gliding model of a soaring bird called the Raven. Hoey's soaring Raven employed a rolling tail similar to that of the MAV. Hoey air launched his model from a conventional radio control aircraft and flew his Raven via rolling tail and drag flap control inputs. His study found that "although the model has no vertical stabilizing or controlling surfaces, it is statically stable and controllable in all axes" (Hoey, 1992:398). He also found that center of gravity,  $cg$ , and roll stability,  $Cl_\beta$ , due to dihedral effect were factors in the successful flights.

The effectiveness of the rolling tail in yawing and turning the model depended on the lift load on the tail in trimmed flight. For a forward  $cg$  the tail was loaded downward. Rotating the tail clockwise produces a left force at the tail and dihedral effect caused a right turn...Several flights

were completed in this configuration using only the rolling tail for pitch and yaw control. (Hoey, 1992:397)

However, the rolling tail configuration had augmented roll and yaw authority from “drag flaps” or spoilers on the wings. There is no specific mention of complete roll control from the rolling tail, but Hoey notes the following:

While in soaring flight Ravens apparently control roll attitude, and thus turning flight, strictly through dihedral effect and the application of a yawing moment to produce sideslip. The yawing moment is typically introduced by tilting the tail but could also be produced by drag differential on the wings. (Hoey, 1992:398)

The unique control inputs required for this type of flight yield handling qualities that are “not very comfortable for a human pilot but they are probably completely normal to a Raven” (Hoey, 1992:397). Current research seeks to better understand the required control inputs and thus bridge the gap between birds and human controlled flight.

### **Multiple Regression**

Previous theses by Deluca, Parga, and Leveron yielded a wealth of data on the flexible wing characteristics and tail effectiveness of the rotatable tail MAV. The unique coupling effect of a single surface, two-axis tail makes the control effect on force and moment coefficients a function of both tail rotation and deflection. The specifics of the coupled effects are detailed in Chapters III and IV, however, the basic statistical methods used to curve-fit the experimental data are described here.

When determining a relationship of two or more variables, it is referred to as a multiple regression model (Kiemele, 1997:7-28). A specific type of multiple regression model that can capture coupling effects is known as a quadratic interaction model. A



second order version of this type of model for a two-variable function has a total of nine coefficients in the form of Equation 1:

$$y(x_1, x_2) = b_0 + b_1x_1 + b_{11}x_1^2 + b_2x_2 + b_{22}x_2^2 + b_{12}x_1x_2 + b_{(11)2}x_1^2x_2 + b_{1(22)}x_1x_2^2 + b_{(11)(22)}x_1^2x_2^2 \quad (1)$$

where  $y$  is the dependent variable,  $x_1$  and  $x_2$  are the independent variables, and each  $b$  term represents the strength of the dependence on the relationship between the two independent variables (Kiemele, 1997:7-32). The  $b$  coefficients of Equation 1 can be determined using the least squares method that minimizes the sum of the squared deviations from the curve (Kiemele, 1997:7-5). This minimization is done in Matlab where “A is an m-by-n matrix with  $m \approx n$  and B is a column vector with m components, or a matrix with several such columns, then  $X = A \backslash B$  is the solution in the least squares sense to the under- or overdetermined system of equations  $AX = B$ ” (Matlab, 2002). One limitation to this method is the inaccuracy of extrapolation beyond the given data set. The combination of interactions modeled within the given data bounds may not similarly apply outside the bounds of the data. The inability to extrapolate affected this research because data were only collected on a small range of tail rotation.

### **Forces and Moments**

There are three forces: lift ( $L$ ), drag ( $D$ ), and sideforce ( $Y$ ); and three moments: roll ( $l$ ), pitch ( $m$ ), and yaw ( $n$ ) about any aircraft. Once the multiple regression models are created from the force and moment coefficient data sets, the values can be estimated for any combination of inputs within the original data set. Each non-dimensional coefficient,  $C$ , has a common form as generalized in Equation 2 (Stevens, 2003:77-97):

$$\begin{aligned}
C_D &= C_D(\alpha, \beta, M, h) + \Delta C_D(M, \delta_e) + \Delta C_D(M, \delta_r) + \Delta C_D(\delta_F) + \Delta C_D(gear) \\
C_L &= C_L(\alpha, \beta, M, T_c) + \Delta C_L(\delta_F) + \Delta C_{L_{ge}}(h) \\
C_Y &= C_Y(\alpha, \beta, M) + \Delta C_{Y_{\delta_e}}(\alpha, \beta, M, \delta_r) + \Delta C_{Y_{\delta_a}}(\alpha, \beta, M, \delta_a) + \frac{b}{2V_T} [C_{Y_p}(\alpha, M)P + C_{Y_r}(\alpha, M)R] \\
C_\ell &= C_\ell(\alpha, \beta, M) + \Delta C_{\ell_{\delta_e}}(\alpha, \beta, M, \delta_r) + \Delta C_{\ell_{\delta_a}}(\alpha, \beta, M, \delta_a) + \frac{b}{2V_T} [C_{\ell_p}(\alpha, M)P + C_{\ell_r}(\alpha, M)R] \\
C_m &= C_m(\alpha, M, h, \delta_F, T_c) + \Delta C_{m_{\delta_e}}(\alpha, M, h, \delta_e) + \frac{\bar{c}}{2V_T} [C_{m_q}Q + C_{m_{\dot{\alpha}}} \dot{\alpha}] + \frac{x_R}{\bar{c}} C_L + \Delta C_{m_{thrust}}(\delta_t, M, h) + \Delta C_{m_{gear}}(h) \\
C_n &= C_n(\alpha, \beta, M, T_c) + \Delta C_{n_{\delta_e}}(\alpha, \beta, M, \delta_r) + \Delta C_{n_{\delta_a}}(\alpha, \beta, M, \delta_a) + \frac{b}{2V_T} [C_{n_p}(\alpha, M)P + C_{n_r}(\alpha, M)R]
\end{aligned} \tag{2}$$

where

$\alpha =$	Angle of attack
$\beta =$	Angle of sideslip
$M =$	Mach number
$h =$	Altitude
$\Delta C =$	Change in coefficient due to input ( )
$\delta_e =$	Elevator deflection
$\delta_r =$	Rudder deflection
$\delta_a =$	Aileron deflection
$\delta_F =$	Flap deflection
$T_c =$	Throttle setting
$b =$	Wing span
$\bar{c} =$	Wing mean geometric chord
$V_T =$	True airspeed
$C_{\ell_p} =$	Roll damping due to roll rate
$P =$	Roll rate (body component)
$Q =$	Pitch rate (body component)
$R =$	Yaw rate (body component)

Many of the terms in Equation 2 are representative of traditional aircraft with rudders, ailerons, flaps, landing gear, and large variations in Mach number that the MAV does not have. The changes in coefficients due to the effect of those types of inputs will not be present in the MAV simulation. Further development of how Equation 2 is applied to this research is found in Chapter III.

Notice the  $C_Y$ ,  $C_b$ ,  $C_m$ , and  $C_n$  terms in Equation 2 all have rate dependent damping terms as the end part of their general form. Damping is the counteracting portion of the individual force or moment of interest. Damping only applies in a dynamic time environment. These damping terms were not measured during static wind tunnel testing done by Deluca, Parga, or Leveron. Therefore, the damping terms must be estimated based on empirical data from previous aircraft. Large volumes of data for existing rigid wing aircraft have produced estimating functions for damping rates. The only damping term modeled by this research is the roll damping due to roll rate term,  $C_{lp}$ . This effect is the result of the wing acting as a damper in fluid air. An illustration of this effect from the Roskam reference is shown in Figure 4.

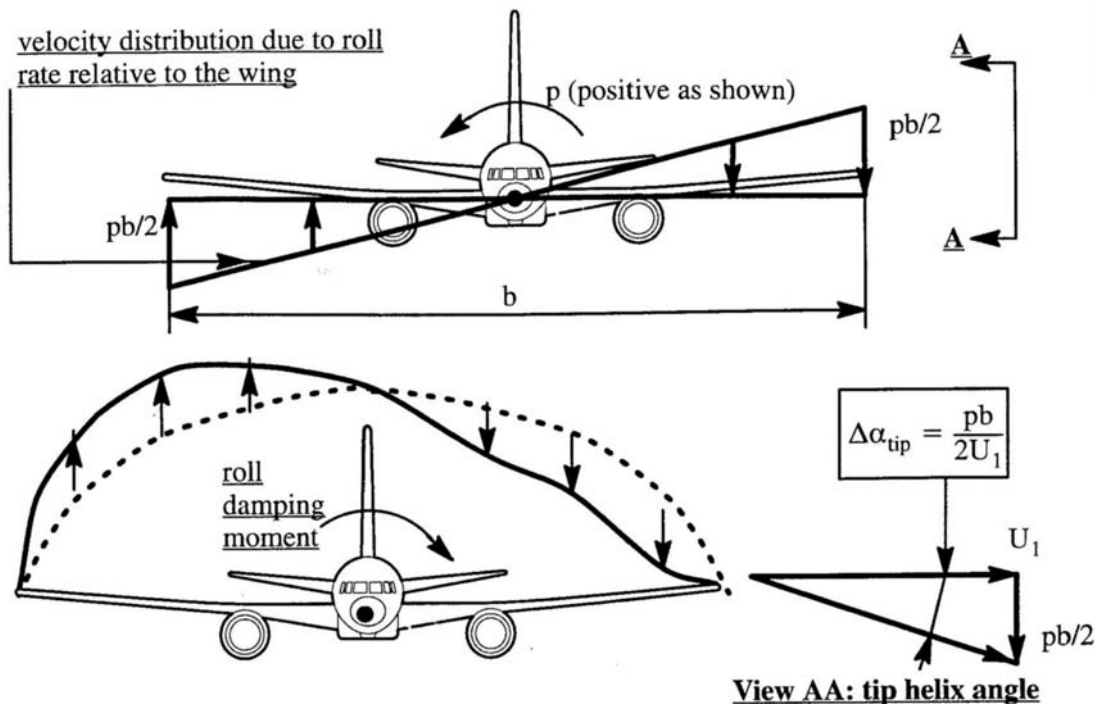


Figure 4. Physical Explanation for Rolling Moment due to Roll Rate (Roskam, 1995:153)

Roskam explains in his text that “as long as the flow remains attached, the effect of perturbed roll rate,  $p$ , is to create an asymmetrical lift distribution which opposes the roll rate” (Roskam, 1995:152-153). Robert Nelson puts forth the following estimate for  $C_{\ell p}$  in Equation 3:

$$C_{\ell p} = -\frac{C_{L\alpha}}{12} \frac{1+3\lambda}{1+\lambda} \quad (3)$$

where  $C_{L\alpha}$  is the airplane lift curve slope and  $\lambda$  is the taper ratio (tip chord/root chord) (Nelson, 1998:121). The MAV uses an efficient elliptical wing planform that technically has a tip chord of zero and therefore no calculable taper ratio so a comparison between tapered and elliptical wings is required. Raymer explains that “a taper ratio of 0.45...produces a lift distribution very close to the elliptical ideal” (Raymer, 1992:56). The tail area is not considered in the roll damping estimate. Therefore, a taper ratio of  $\lambda = 1$  was used in this research to develop 19% more damping from the Equation 3 estimate than the  $\lambda = 0.45$  of a true elliptical wing. Improved roll damping performance should help stop oscillations caused by perturbations in roll.

The University of Florida’s aeroelastic fixed wing micro aerial vehicle uses a similar construction, but with a shorter wingspan than the MAV. Waszak, Jenkins, and Ifju (2001) explain “the passive mechanism of adaptive washout” in their paper as an “extension of the membrane and twisting of the structure in response to changes in speed and vehicle attitude causing changes in angle of attack along the span. The effect is to reduce the response of the vehicle to disturbances” (Waszak, Jenkins, and Ifju, 2001).

While their study does not specifically address roll damping due to roll rate, it suggests that a flexible wing has notably more inherent gust damping properties than a rigid wing.

The Waszak, Davidson, and Ifju (2002) paper on simulation and flight control of the aforementioned vehicle focuses more specifically on roll, pitch, and yaw rate control. Their paper states that “additional terms were added to the Taylor series in an ad hoc manner to account for dependence on angular rates...these terms were computed using PMARC” which is a potential flow theory based panel code (Waszak, 2002;| Ashby, 1999). Values from their study are not directly transferable to the MAV due to drastically different dimensions—specifically span and aspect ratio.

Coefficient equations including both the static stability derivatives and the dynamically dependent damping terms are input to the three force and three moment equations. Equation 4 lists all six equations in terms of “wind-axes components” where  $\alpha$  is the angle of attack and  $\beta$  is the sideslip angle (Stevens, 2003:72, 76):

$$\begin{aligned}\bar{q} &= \frac{1}{2} \rho V^2 \\ D &= \bar{q} S C_D \\ L &= \bar{q} S C_L \\ Y &= \bar{q} S C_Y \\ \ell &= \bar{q} S b C_\ell \\ m &= \bar{q} S \bar{c} C_m \\ n &= \bar{q} S b C_n\end{aligned}\tag{4}$$

where  $\bar{q}$  is dynamic pressure,  $\rho$  is the density of air,  $V$  is the true airspeed,  $S$  is the planform area, and  $C$  is the corresponding coefficient. Stevens also notes the following about alpha and beta dependence:

As a rough generality, longitudinal coefficients (lift, drag, pitching moment) are primarily dependent on alpha, and in the lateral-direction coefficients (roll, yaw, and sideforce) beta is equally as important as alpha. (2003:76)

The mathematical model for the forces and moments about the aircraft is used to determine a balanced combination of control inputs and states for a desired trimmed flight condition.

### **Optimization**

Numerical optimization is a common practice for identifying trimmed conditions. Bryson explains that static optimization “involves finding the values of  $p$  parameters  $y_1, \dots, y_p$  that minimize a performance index that is a function of these parameters,  $L(y_1, \dots, y_p)$ ” (Bryson, 1999:1). The performance index,  $L$ , is also referred to as the cost function of the optimization. The cost function is dependent on parameters  $y_1, \dots, y_p$  known as the Design Variables (DVs). Static optimization finds the minimized cost function for a single point in time. The general static optimization problem includes equality and inequality constraints for the values of the parameters  $y_1, \dots, y_p$ . These constraints limit the range of values for the DVs.

One powerful numerical algorithm for optimization problems with constraints is Matlab’s *fmincon* function in the Optimization Toolbox (Matlab, 2002). The *fmincon* function accepts user defined cost and constraint functions along with initial estimate design variable values. This Matlab function is used to find appropriate control inputs

and initial states to balance the forces and moments about an aircraft for trimmed flight. Further development of the specific cost function and constraints used in this research are found in Chapter III.

### Non-linear Equations of Motion

After the optimal variables are determined for trimmed flight, the highly non-linear dynamic behavior of the aircraft is modeled using the full non-linear Equations Of Motion (EOMs). Often, a simplified linear set of equations of motion are used for aircraft modeling. However, the coupled control effects of the rotatable tail in this unique configuration of the MAV make the complete non-linear EOMs a necessity. Basic assumptions for the EOMs include “earth is an inertial reference frame, aircraft mass properties are constant, and the aircraft has a plane of symmetry” (Honeywell, 1996:61). Two common systems of state variables to describe dynamic motion are body components and flight-path components summarized in Table 2 (Honeywell, 1996:61).

Table 2. Choices for State Variables (Honeywell, 1996:61)

Body Components		Flight Path Components	
Variable	Symbol	Variable	Symbol
Roll Rate	$p$	Roll Rate	$p$
Pitch Rate	$q$	Pitch Rate	$q$
Yaw Rate	$r$	Yaw Rate	$r$
Longitudinal Velocity	$u$	Velocity	$V$
Lateral Velocity	$v$	Sideslip Angle	$\beta$
Normal Velocity	$w$	Angle of Attack	$\alpha$
Euler Roll Angle	$\phi$	Bank Angle (about velocity vector)	$\mu$
Euler Pitch Angle	$\theta$	Flight-Path Angle	$\gamma$
Euler Yaw Angle	$\psi$	Heading Angle	$\chi$
North Position	$\xi$	North Position	$\xi$
East Position	$\eta$	East Position	$\eta$
Altitude	$h$	Altitude	$h$

The aircraft body axes positive values are forward, out the right wing, and down for  $x$ ,  $y$ , and  $z$  respectively (Honeywell, 1996:61). The flight-path component EOMs are useful for flight path description and visualization and are used throughout the remainder of this research. Honeywell shows that the “flight-path state variables can be derived from the body state variables as follows” in Equation 5 (1996:61):

$$\begin{aligned}
 V &= \sqrt{u^2 + v^2 + w^2} \\
 \beta &= \sin^{-1} \left( \frac{v}{\sqrt{u^2 + v^2 + w^2}} \right) \\
 \alpha &= \tan^{-1} \left( \frac{w}{u} \right) \\
 \mu &= \tan^{-1} \left( \frac{uv \sin \theta + (u^2 + w^2) \sin \phi \cos \theta - vw \cos \phi \cos \theta}{\sqrt{u^2 + v^2 + w^2} (w \sin \theta + u \cos \phi \cos \theta)} \right) \\
 \gamma &= \sin^{-1} \left( \frac{u \sin \theta - v \sin \phi \cos \theta - w \cos \phi \cos \theta}{\sqrt{u^2 + v^2 + w^2}} \right) \\
 \chi &= \tan^{-1} \left( \frac{u \cos \theta \sin \psi + v(\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) + w(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)}{u \cos \theta \cos \psi + v(\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) + w(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)} \right)
 \end{aligned} \tag{5}$$

Develop the EOMs by letting “ $T$  denote the propulsion force along the  $\hat{b}_x$  direction and  $\ell_p^c$ ,  $m_p^c$ ,  $n_p^c$  propulsion moments for roll, pitch, and yaw, respectively” (Honeywell, 1996:62). The full set of 12 non-linear EOMs using flight-path components are set forth by Honeywell in Equation 6 (1996:65):



$$\begin{aligned}
I_{xx}\dot{p} - I_{xz}\dot{r} &= \ell_a^c + \ell_p^c + I_{yy}rq + I_{xz}pq - I_{zz}qr \\
I_{yy}\dot{q} &= m_a^c + m_p^c - I_{xz}p^2 + I_{zz}pr - I_{xx}rp + I_{xz}r^2 \\
- I_{xz}\dot{p} + I_{zz}\dot{r} &= n_a^c + n_p^c + I_{xx}pq - I_{xz}qr - I_{yy}pq \\
\dot{V} &= \frac{1}{m}(-D \cos \beta + Y \sin \beta + T \cos \beta \cos \alpha) - g \sin \gamma \\
\dot{\chi} &= \frac{1}{mV \cos \gamma} [D \sin \beta \cos \mu + Y \cos \mu \cos \beta + L \sin \mu + T(\sin \mu \sin \alpha - \cos \mu \sin \beta \cos \alpha)] \\
\dot{\gamma} &= \frac{1}{mV} [-D \sin \beta \sin \mu - Y \sin \mu \cos \beta + L \cos \mu + T(\cos \mu \sin \alpha + \sin \mu \sin \beta \cos \alpha)] - \frac{g \cos \gamma}{V} \\
\dot{\mu} &= \frac{p \cos \alpha + r \sin \alpha}{\cos \beta} + \frac{1}{mV} [D \sin \beta \cos \mu \tan \gamma + Y \tan \gamma \cos \mu \cos \beta + L(\tan \beta + \tan \gamma \sin \mu) \\
&\quad + T(\sin \alpha \tan \gamma \sin \mu + \sin \alpha \tan \beta - \cos \alpha \tan \gamma \cos \mu \sin \beta)] - \frac{g \cos \gamma \cos \mu \tan \beta}{V} \\
\dot{\alpha} &= q - \tan \beta (p \cos \alpha + r \sin \alpha) - \frac{1}{mV \cos \beta} (L + T \sin \alpha) + \frac{g \cos \gamma \cos \mu}{V \cos \beta} \\
\dot{\beta} &= -r \cos \alpha + p \sin \alpha + \frac{1}{mV} (D \sin \beta + Y \cos \beta - T \sin \beta \cos \alpha) + \frac{g \cos \gamma \sin \mu}{V} \\
\dot{\xi} &= V \cos \gamma \cos \chi \\
\dot{\eta} &= V \cos \gamma \sin \chi \\
\dot{h} &= V \sin \gamma
\end{aligned} \tag{6}$$

## Non-linear Solver

The coefficients of Equation 2 feed into the forces and moments of Equation 4 and those values are used in the EOMs of Equation 6. All of these equations must be solved simultaneously at each time step in a dynamic simulation. A complicated system of non-linear equations like those of Equations 2, 4, and 6 require advanced numerical techniques for solution. The algorithm employed for this research is Matlab's ode45 function. This function uses the current state vector along with the force and moment

coefficients to numerically solve the 12 ordinary differential equations (ODEs) that form the non-linear EOMs (Matlab, 2002). The algorithm “is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair. It is a one-step solver—in computing  $y(t_n)$ , it needs only the solution at the immediately preceding time point,  $y(t_{n-1})$ ” (Matlab, 2002). Ode45 is a variable time-step method with bounds set by the user or left in auto mode for the algorithm to determine. The Ode45 function is imbedded within a Simulink model that integrates each time step to propagate the states forward in time. A full description of the specific models used follows in Chapter III.

### **Linearization and Open Loop Stability**

One way to analyze complicated non-linear systems is to linearize the non-linear system, find its eigenvalues, and evaluate the stability. Matlab’s linmod function can “extract the linear state-space model of a system around an operating point” (Matlab, 2002). This tool takes the initial input,  $u_0$ , and state,  $x_0$ , values, perturbs the non-linear Simulink model to obtain numerical partial derivatives, and creates the linear  $A$ ,  $B$ ,  $C$ , and  $D$  matrices used in Equation 7 to describe the input-output relationship:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{7}$$

The resulting  $A$  matrix is of primary interest for stability analysis. If the states are ordered properly, the  $A$  matrix can be broken down into sub matrices specific to longitudinal and lateral-directional modes. This technique is referred to as “partitioning” by Stevens and Lewis. The proper state ordering for phugoid, short period, roll, and

dutch roll partitioning based on the previously discussed flight-path state variables is listed in Equation 8 (Stevens, 2003:263-274):

$$x = \begin{bmatrix} V_T \\ \gamma \\ \alpha \\ q \\ p \\ \mu \\ \beta \\ r \\ \chi \\ \xi \\ \eta \\ h \end{bmatrix} \quad (8)$$

Note there are two state variables for each of the modes of interest. The last four state variables are not included in the specific ordering, but are necessary for solution of the 12 EOMs.

The real part of the eigenvalues for the phugoid, short period, roll, and dutch roll modes need to be negative for stability. The short period is usually a much larger negative value than the phugoid (Stevens, 2003:210-211). If these characteristics are not naturally present in the basic design of the aircraft, Stability Augmentation Systems may be used to achieve the desired stability.

The theoretical background for each piece of the research flowchart has been laid out in this chapter. Each part of the process builds upon the previous to reach the goal of simulating dynamic aircraft behavior based on experimental wind tunnel data. The

specific methods used to apply all of the theory from this literature review are discussed in Chapter III.

### **III. Methodology**

#### **Chapter Overview**

The purpose of this chapter is to detail the specific approach used for characterizing and simulating the flexible wing, rotatable tail MAV. The research approach is the application of the principals and theory from the literature review in Chapter II. First, the existing experimental data is curve fit to create mathematical models for the force and moment coefficients. Then, the forces and moments are calculated including the dynamic damping terms that were not measured in previous experiments. A static optimization systematically perturbs the control inputs and states to balance the forces and moments as close to zero as possible for trimmed Steady Level Unaccelerated Flight (SLUF). The trimmed control inputs and states are used as the initial conditions for the Matlab/Simulink non-linear EOM solver. The solver calculates the change in each of the 12 state variables at each time step and integrates them over time.

Once the EOMs are built up, different options can be utilized to investigate stability, controllability, and achievable performance. The first option is to perform a linearization of the model to analyze the longitudinal and lateral-directional stability of the aircraft, which can be done by studying the eigenvalues of the linear model. Furthermore, the linear model can be used to develop a linear feedback controller to improve the stability or even to keep the non-linear system in SLUF. Another option is the use of an optimization in the loop of the simulation that attempts to find the

appropriate tail position at each time step to keep the aircraft as close to its desired state as possible. This whole process is described in depth in the paragraphs that follow.

### Curve Fits

The experimental wind tunnel data from Deluca, Parga, and Leveron were used to create mathematical models for the force and moment coefficients of the MAV as a function of the angle of attack,  $\alpha$ , and angle of sideslip,  $\beta$ , as well as the tail deflection,  $\delta_e$ , and tail rotation  $\delta_m$ . Note that the MAV does not have the traditional aileron,  $\delta_a$ , or rudder,  $\delta_r$ , deflection terms found in Equation 2 because of its unique tail and lack of roll control mechanisms on the wing. However, the MAV keeps the traditional  $\delta_e$  term that is directly coupled with  $\delta_m$  term. The sign convention used for  $\delta_e$  and  $\delta_m$  is consistent with that of Parga's and Leveron's research, and is illustrated in Figure 5. Tail up is negative  $\delta_e$  and clockwise rotation of the tail from the aft view of the aircraft is positive  $\delta_m$ .

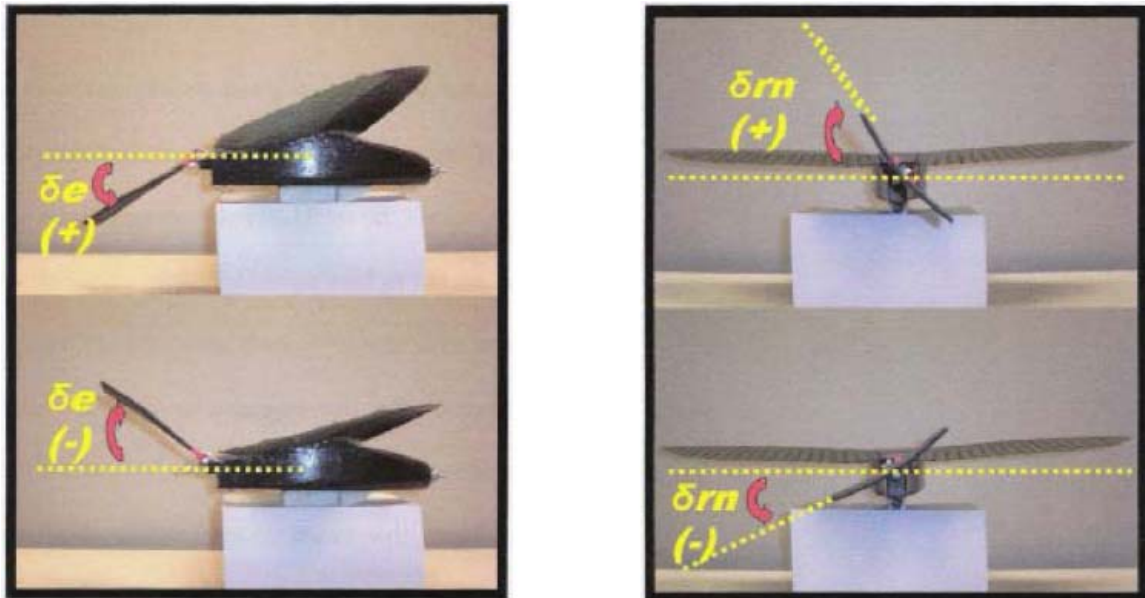


Figure 5. Tail Deflection and Rotation Sign Convention (Parga, 2004:59)

The primary goal of the curve fitting process is to capture both the aircraft stability derivative without tail deflections as a function of  $\alpha$  or  $\beta$  and the tail effect on the coefficients when in a non-zero position. The general set of equations for the force and moment coefficients shown in Equation 2 is applied to this research as Equation 9.

$$\begin{aligned}
C_D &= C_D(\alpha) + \Delta C_D(\delta_m, \delta_e) \\
C_L &= C_L(\alpha) + \Delta C_L(\delta_m, \delta_e) \\
C_m &= C_m(\alpha) + \Delta C_{m_{\delta_e}}(\delta_m, \delta_e) \\
C_\ell &= C_\ell(\beta) + \Delta C_{\ell_{\delta_r}}(\delta_m, \delta_e) + C_{\ell_p} p \\
C_n &= C_n(\beta) + \Delta C_{n_{\delta_r}}(\delta_m, \delta_e) \\
C_Y &= C_Y(\beta) + \Delta C_{Y_{\delta_r}}(\delta_m, \delta_e)
\end{aligned} \tag{9}$$

First, the coefficients are segregated by their primary dependence on either  $\alpha$  or  $\beta$ . The  $\alpha$ -dependent coefficients are  $C_D$ ,  $C_L$ , and  $C_m$ . The  $\beta$  dependent coefficients are  $C_\ell$ ,  $C_n$ , and  $C_Y$ . An argument can be made that some of these are dependent on both  $\alpha$  and  $\beta$ —especially the roll moment coefficient. In fact,  $C_l(\alpha, \beta)$  was initially modeled in this research, but later simplified to only a  $\beta$  dependence in an effort to focus on the tail effectiveness mapping. All of the  $\alpha$  and  $\beta$  fits are generated by the Microsoft Excel curve fitting tool.

The  $\alpha$ -dependent coefficients are based on Parga's data where  $\beta=0.26$ ,  $\delta_e=0$  and  $\delta_m=0$  degrees and  $\alpha$  was swept from -8.249 to +10.806 degrees. The  $C_D(\alpha)$ ,  $C_L(\alpha)$ , and  $C_m(\alpha)$  fits were matched by third, third, and fourth order polynomial fits respectively to achieve an  $R^2$  value of 0.998 or better. The  $\beta$ -dependent coefficients are based on Levern's data where  $\alpha=4.8$ ,  $\delta_e=0$  and  $\delta_m=0$  degrees and  $\beta$  was swept from -8 to +8

degrees. The  $C_l(\beta)$ ,  $C_n(\beta)$ , and  $C_Y(\beta)$  fits were matched by linear curve fits to achieve an  $R^2$  value of 0.992 or better. All data were collected with a wind tunnel freestream speed of approximately 30 miles per hour (mph).

All tail effectiveness data used for this research came from Leveron's thesis because it included the widest range of tail deflections and rotations and was representative of the most current configuration to include the foldable rotatable tail. The specific sub-set of his data used is the Tail 1 Stab 1 data that measured the effect of the flat tail on the MAV configured with aft ventral fins for improved yaw stability. This data was chosen in hopes for the greatest chance of success in achieving stable trimmed flight. All code and procedures could be applied to the other cases analyzed in Leveron's work. The original data from Leveron's research was copied and resorted into ascending order for  $\delta_m$  and ascending order for  $\delta_e$  within the sorted  $\delta_m$  groups and is found in Appendix A. Then, the data files were saved as .prn files to be loaded by Matlab for fitting. There are six separate files found in Appendix B—one for each coefficient—that reference the same .prn data file for curve fitting. The structure of each of those files is the same and is described in the following paragraphs.

First, the columns of interest within the data file are loaded. This includes the  $\alpha$ ,  $\beta$ ,  $\delta_m$ , and  $\delta_e$  values and the corresponding coefficient values for the specific file. Remembering this is experimental and not ideal or theoretical data, there is inherent bias and uncertainty within each measurement. To simplify the modeling process, coefficients that should theoretically cross at zero,  $C_l(\beta)$ ,  $C_n(\beta)$ , and  $C_Y(\beta)$ , have a correction applied that the other coefficients do not. The bias at the  $\delta_e=0$  and  $\delta_m=0$



degrees point is identified and removed from all of the measured coefficient values in that column.

The  $\alpha$  and  $\beta$  dependence with no tail deflection or rotation is included in the total measured coefficients of the tail effectiveness data. The contribution due to either  $\alpha$  or  $\beta$  is calculated for each test point and subtracted from the total measured value prior to curve fitting the contribution of the coupled  $\delta_e$  and  $\delta_m$  settings. A trigonometric combination was first considered for the curve fit, but would have led to potential singularities and control law difficulties as the research progressed. Because the remaining coefficient value is a function of both  $\delta_e$  and  $\delta_m$ , the multiple regression technique described in Chapter 2 is necessary. The second order quadratic interaction model of Equation 1 is applied to the specific coefficient case and takes the form of Equation 10:

$$C_{( )}(\delta_m, \delta_e) = C_{( )}(\alpha \text{ or } \beta) + x_1 + x_2\delta_m + x_3\delta_m^2 + x_4\delta_e + x_5\delta_e^2 + x_6\delta_m\delta_e + x_7\delta_m^2\delta_e + x_8\delta_m\delta_e^2 + x_9\delta_m^2\delta_e^2 \quad (10)$$

All of the fitting is done using Matlab's matrix math capabilities. The  $x_1$  to  $x_9$  fit terms are found by forming vectors  $A_1$  to  $A_9$  with length equal to the number of data points and stacking them into matrix  $A$  of the form in Equation 11 and applying Matlab's least squares operator "\" as shown in Equation 12:

$$\begin{aligned}
A_1 &= 1 \\
A_2 &= \delta_m \\
A_3 &= \delta_m^2 \\
A_4 &= \delta_e \\
A_5 &= \delta_e^2 \\
A_6 &= \delta_m \delta_e \\
A_7 &= \delta_m^2 \delta_e \\
A_8 &= \delta_m \delta_e^2 \\
A_9 &= \delta_m^2 \delta_e^2 \\
A &= [A_1 \quad A_2 \quad A_3 \quad A_4 \quad A_5 \quad A_6 \quad A_7 \quad A_8 \quad A_9]
\end{aligned} \tag{11}$$

$$x = A \setminus (C_{(raw)} - C_{(bias)} - C_{(\alpha \text{ or } \beta)}) \tag{12}$$

The calculated fit terms  $x_1$  to  $x_9$  are saved to file for later use in trimming and simulation. Then the raw data is compared to the curve fit model in a side-by-side comparison of colorized contour and surface plots shown in Chapter IV. It is important to remember from Chapter II that these curve fits are only valid within the bounds of the experimental data that was taken. When using these math models to calculate the forces and moments, the non-linearity of the fit functions invalidates extrapolated values outside the original data range.

### **Forces and Moments**

The three force and three moments about the MAV are calculated in a stand-alone Matlab file named *ForcesMoments.m* found in Appendix C that can be called any time during the simulation when the forces and moments are needed. This function uses the curve fit terms,  $x_1$  to  $x_9$  to form the force and moment coefficients. It also uses the states

that are passed to it to solve Equation 4. The only damping term applied to the forces and moments in this research is the roll moment damping due to roll rate,  $Cl_p$ . This term was discussed in Chapter II and estimation for this term shown in Equation 3 with an intentionally overestimated  $\lambda=1$  to account for extra damping due to the flexible wing and tail surface.  $Cl_p$  is the only damping term added because most other significant damping terms are dependent primarily on vertical surfaces which are small on the latest modification, and not part of the original rotatable tail design. Pitch damping terms were not considered in this study because the lateral-directional modes were most problematic and drew the majority of the focus. The concept of improved damping characteristics from the flexible wing led to the notion of scaling  $Cl_p$  for a parametric sensitivity study of how roll damping behavior may affect SLUF performance and handling qualities. The results of varying  $Cl_p$  as well as  $Cn_\beta$  and  $Cl_\beta$  will be discussed in Chapter IV.

### **Trimmer**

Seeking to achieve SLUF conditions first involves calculating the trimmed control inputs and states required to sum the forces and moments about the aircraft to zero. The highly coupled nature of the rotatable tail makes trimmed flight a tricky balancing act. Static optimization was chosen to perturb inputs and states within defined constraints to minimize the cost function.

The trimmer code used is a separate Matlab file found in Appendix D. It loads the force and moment coefficient models from file and passes Matlab's *fmincon* function an initial guess for trimmed SLUF. This function is capable of choosing either an angle of attack or an airspeed for trimmed flight and determining the optimal parameters for all

other states. However, all research was done using  $V = 30$  feet per second (ft/s) or 20.5 mph as the desired trim speed. This trim speed was chosen because of mission requirements for small scale surveillance. The initial altitude was selected at 50 ft above sea level, and the air density,  $\rho$ , used is the standard day sea level value of 0.002378 slugs/ft<sup>3</sup>. The physical aircraft properties used for all simulation runs are listed in Table 1 found in Chapter I.

The moments of inertia are converted from  $lb_m in^2$  calculations for a similar MAV provided by AFRL/MNAV. Unit conversion calculations were the only adjustment applied to the given moment of inertia data. All other values are the stated values from Leveron's thesis.

The constraints for the control inputs and state variables are based on natural physical limitations and data ranges from previous research. All of the values except for the true airspeed,  $V_T$ , were created as inequality constraints that allow them to fall within a desired band. The thrust,  $T$ , input must be greater than or equal to zero. It was left to the operator to determine if the optimized  $T$  was beyond reasonable. Leaving the freedom for a more optimized  $T$  could help identify a high drag state that could then be re-evaluated. The tail inputs were first constrained to a region of  $-30^\circ \leq \delta_e \leq +15^\circ$  and  $-20^\circ \leq \delta_m \leq +20^\circ$  based on original data ranges. Later trimmer runs fixed the tail rotation angle at zero degrees to simplify the starting conditions for the open loop simulation. Angle of attack,  $\alpha$ , is the only state variable with a constrained band of available values of  $-4^\circ \leq \alpha \leq +12^\circ$ . This range matches the alpha range of previous data collection. The angle of sideslip,  $\beta$ , is constrained to equal zero during the trimming optimization. All of

the rates about the aircraft are set to zero in the initial optimized estimate with an associated penalty in the cost if they become greater than zero.

The cost function for the trimmer optimizer is based on the concept of zeroing the sum of the forces and moments about the aircraft for SLUF. The two main equations within the cost are for lift verses weight and thrust verses drag. The remaining pieces of the cost function are simply the absolute values of sideforce, pitch, yaw, and roll moments. Since the aircraft does not have ailerons, the roll moment is given added weight (three times that of the other moments) in the cost. The total cost function is shown in Equation 13:

$$J = K_1(L + T \sin(\alpha) - W)^2 + K_2(T \cos(\alpha) - D)^2 + K_3|Y| + (K_4|m| + K_5|n| + K_6|l|) \frac{1}{L_{ch}} \quad (13)$$

Where  $K_i$  = non-dimensional scale and  $L_{ch}$ =characteristic length and

$$K_1 = K_2 = K_3 = 1$$

$$K_4 = K_5 = L_{ch} = 1$$

$$K_6 = 3L_{ch}$$

*Fmincon* uses the given cost constraint functions to perturb the DVs ( $T$ ,  $\delta_e$ ,  $\delta_{rn}$ , and  $\alpha$ ), calculate the resulting forces and moments, and evaluate the corresponding cost. It continues to perturb the DVs until it reaches a minimum value for the cost. Once the minimized cost is determined, the full set of input and state initial conditions are passed to the simulator. The scale value,  $K_6$ , on the roll moment in Equation 13 was given triple weight because it originally had the highest non-zero trimmed value. The residual roll moment was almost immediately diverging the aircraft from SLUF, so added weight in

the cost function forced the trim condition to a lower initial roll rate at a machine zero value.

## EOM Solver

The dynamic open loop simulation is the integration of the EOM solution at each time step. All 12 flight-path EOMs are resident in the BATCAMsim.m Matlab file included in Appendix E. This file is passed the input and state variables by the Simulink model each iteration of the loop until the designated simulation time elapses or the altitude of the aircraft,  $h$ , is less than or equal to zero. The open-loop Simulink model used for this research is shown in Figure 6.

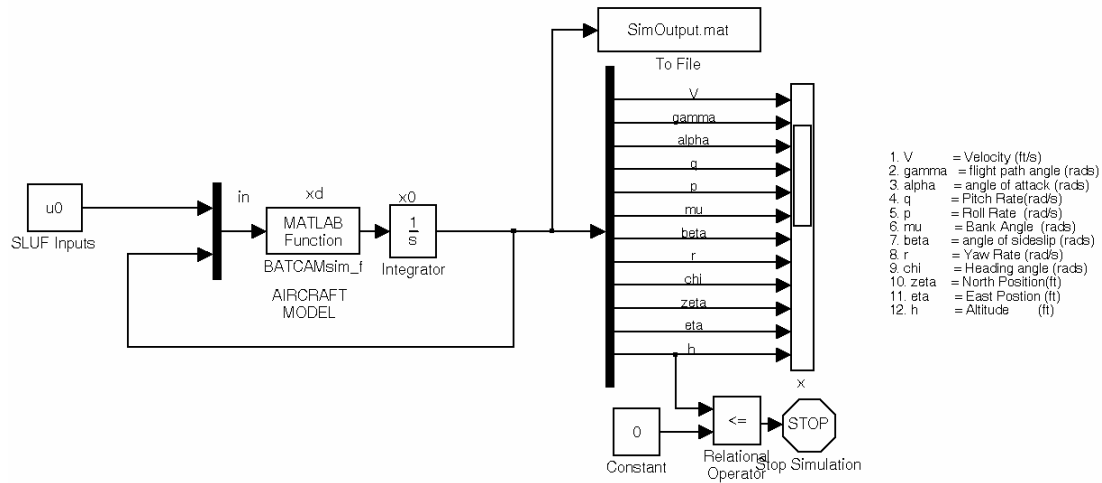


Figure 6. Dynamic Open Loop Simulink Model.

The control inputs,  $u_0$ , are constant during this simulation and represent keeping the thrust, tail deflection and tail rotation at the same setting and releasing the aircraft at the pre-determined initial state conditions. The initial state conditions,  $x_0$ , are resident in the integrator block, and are fed around the loop on the first iteration. All following

iterations have new modified states due to the solution and integration of the EOMs. The states from each time step are saved along with their corresponding run time to the SimOutput.mat file. This file is later loaded and used by the PlotOutput.m file found in Appendix F.

The PlotOutput.m file plots a top, aft, and side view of the flight path as well as the initial conditions used in the simulation. It also plots a three-dimensional positional flight path, all 12 states on a strip chart, and zooms in on crucial states from the simulation. All of the axes and plots can be changed as needed within the PlotOutput.m file to adapt to output analysis needs.

The open loop simulator was verified using aircraft physical properties and stability coefficients from a small conventional rubber band powered aircraft called the Pirol. Both SLUF and steady level turns were tested with matching results to previous research. This provided confidence in results for the MAV values. The results of the MAV open loop sims will be discussed in detail in Chapter IV, but it is worth mentioning that the MAV did not successfully exhibit SLUF after being trimmed. This led to further investigation of a linearized system.

## Linearization

The linearization technique described in Chapter II was applied to the non-linear open loop simulation. The BATCAMlin.m file in Appendix G uses Matlab linmod command to call the Simulink model and calculate linear state space  $A$ ,  $B$ ,  $C$ , and  $D$  matrices. The  $A$  matrix from the state space system is of primary concern. It is partitioned into sub-matrices for phugoid, short period, roll, and dutch roll. The eigenvalues of each mode are calculated.

The state space system and the modal eigenvalues are saved as a tab delimited ascii file named BATCAM\_linmod for later formatting and analysis in Excel. The BATCAM\_linmod file is imported to Excel via tab delimiting, and saved as a new sheet within BATCAM\_linmod.xls. The BATCAM\_linmod\_pretty.xls file shown in Appendix H references the basic numerical data in the BATCAM\_linmod.xls file, and formats it into an interpretable breakdown. Analysis of the eigenvalues yield clues to undesired aircraft behavior and areas that may need improved static stability derivatives or some form of stability augmentation.

The overall structure of the applied theory and background of this research has been set forth, and described in detail. Many challenging steps must build upon one another to yield the desired flight simulation results. The results and analysis of the methodology in this chapter are presented and discussed in Chapter IV.



## IV. Results and Analysis

### Chapter Overview

This chapter describes and analyzes the results of the applied theory and methodology described in Chapters II and III. Overall results include the predicted highly non-linear saddle shaped curve fits for the lateral-directional coefficients, small angle of attack and tail deflection for the initial trimmed conditions, and inherently unstable open loop dynamic behavior—especially in roll and yaw control. Several attempts to improve the SLUF open loop behavior included linear feedback on the non-linear system, in-the-loop optimization for control effectiveness evaluation, and a parametric analysis of artificially improved directional stability and roll damping. Each separate category of the research is described in detail in the following text.

### Curve Fits

The first step in simulating the dynamic behavior of the MAV was developing mathematical models for the force and moment coefficients ( $C_D$ ,  $C_L$ ,  $C_Y$ ,  $C_b$ ,  $C_m$ , and  $C_n$ ) from the previously obtained experimental data. Two sets of curve fits were required for each coefficient—one with zero tail deflections and zero tail rotations with varied  $\alpha$  or  $\beta$ , and one for tail effectiveness based on varied  $\delta_m$  and  $\delta_e$ . Figures 7 through 9 are the  $\alpha$ -dependent coefficient curves based on Parga's thesis data. The polynomial fits are red dashed lines with their corresponding equations and  $R^2$  values displayed.

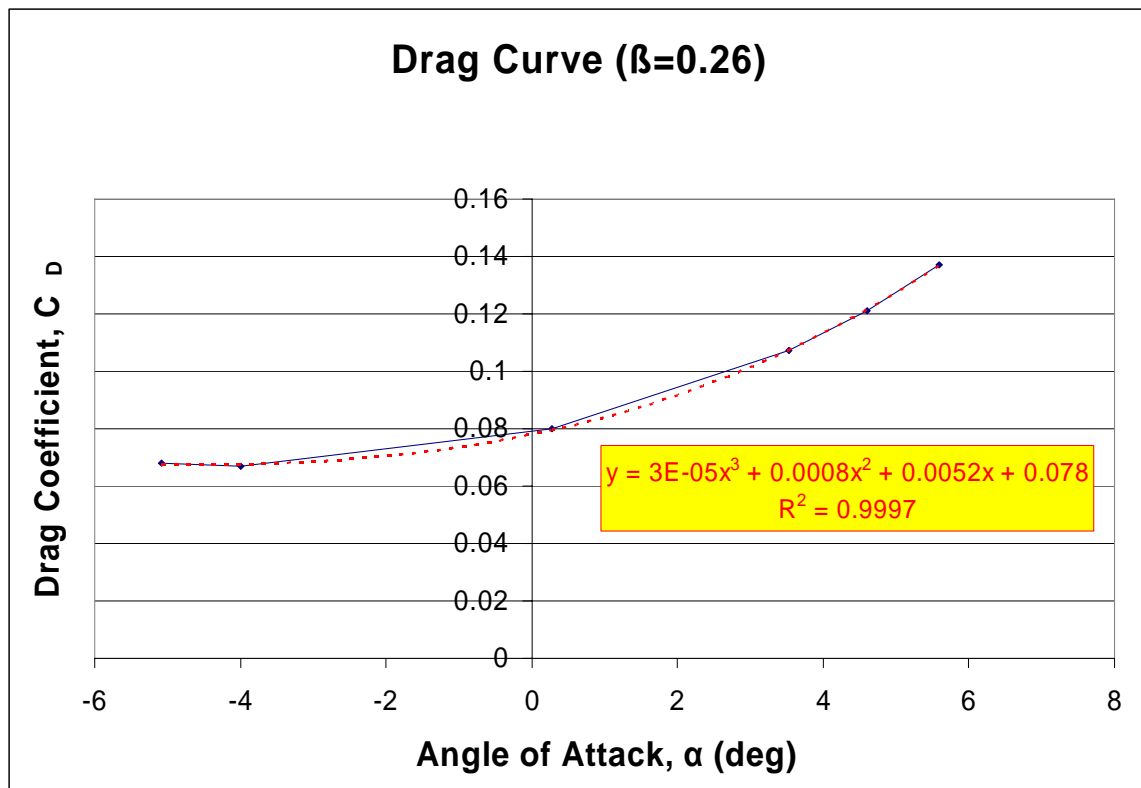


Figure 7.  $C_{D\alpha}$  for Zero Tail Inputs (Parga, 2004)

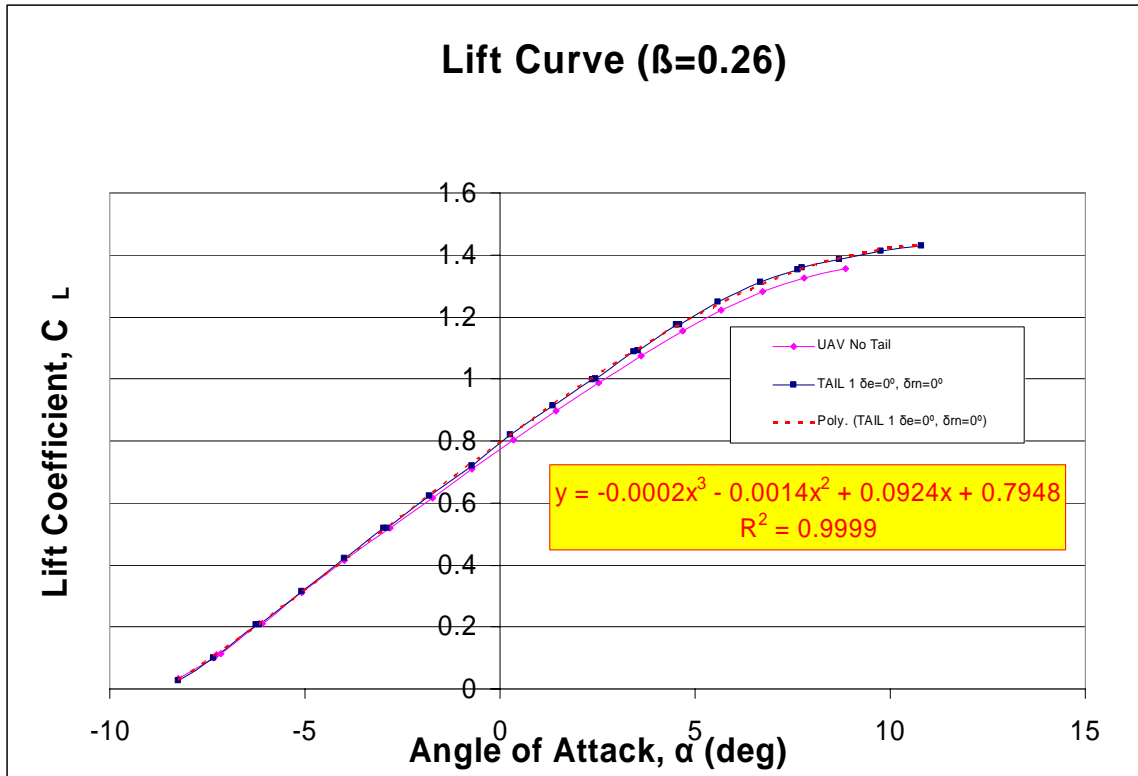


Figure 8.  $C_{L\alpha}$  for Zero Tail Inputs (Parga, 2004)

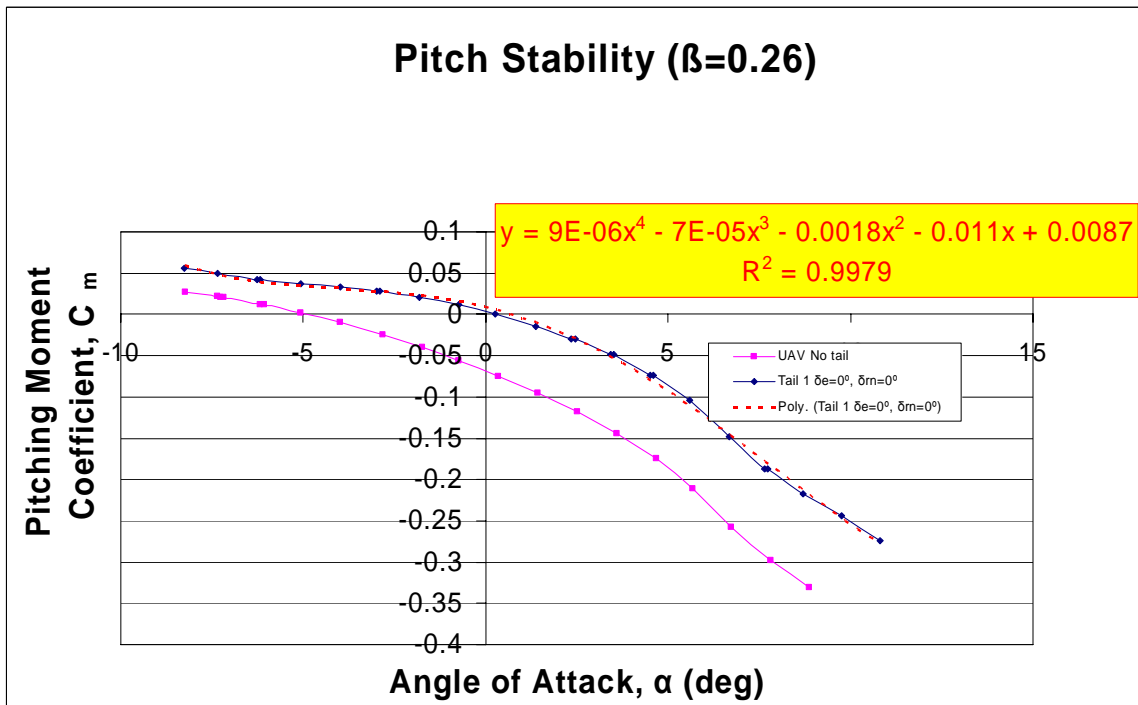


Figure 9.  $C_{m\alpha}$  for Zero Tail Inputs (Parga, 2004)

All of the  $\alpha$  dominated curves meet standard stability criteria. Figures 10 through 12 are the  $\beta$ -dependent coefficient curves based on Leveron's thesis data. The linear fits are shown with red dashed lines and their corresponding equations with  $R^2$  values.

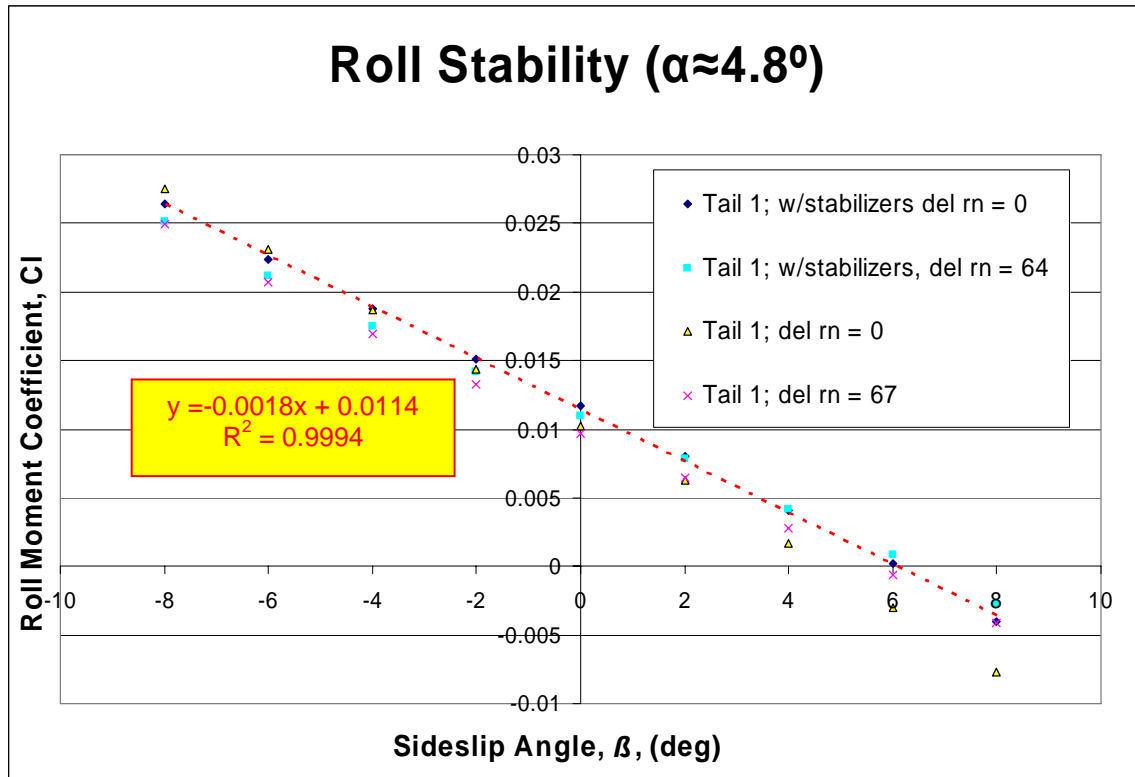


Figure 10.  $C_{l\beta}$  for Zero Tail Inputs (Leveron, 2005)

The curve fit yields a  $C_{l\beta}$  of -0.0018 that meets static stability requirements.

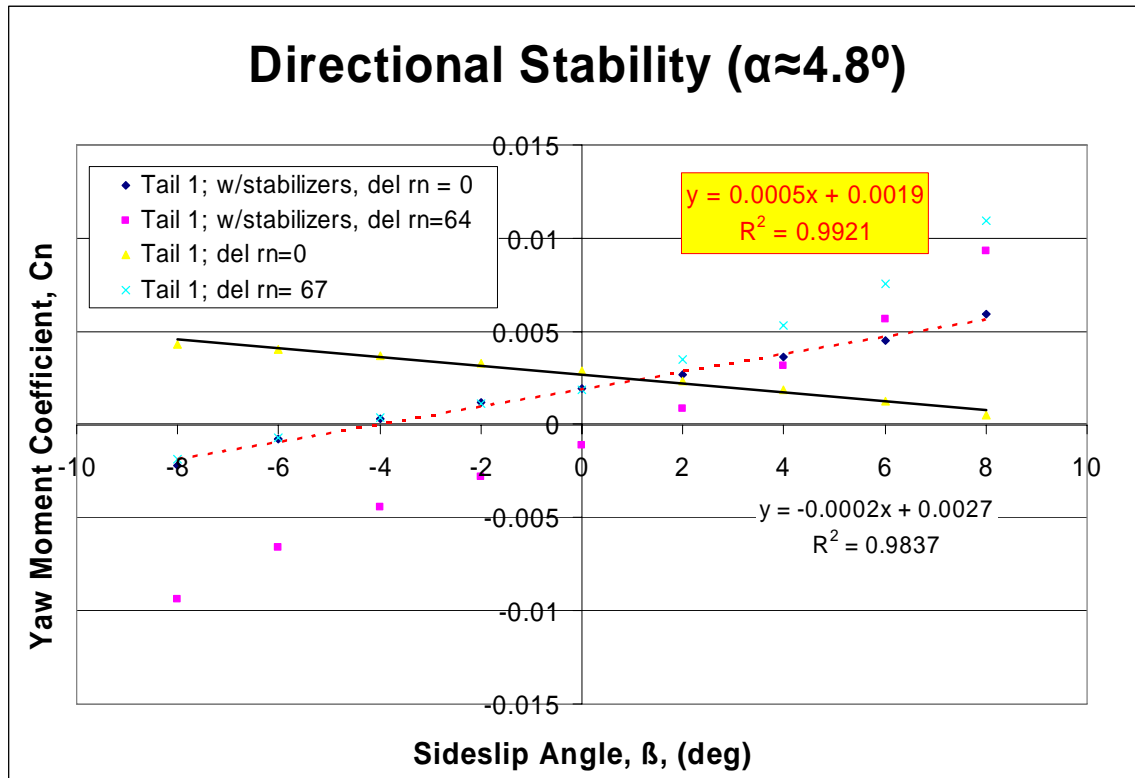


Figure 11.  $C_{n\beta}$  for Zero Tail Inputs (Leveron, 2005)

The positive slope of the configuration with stabilizers indicates very light directional stability that may approach marginal tendencies during flight test. The second linear fit in Figure 11 is the original foldable tail configuration without aft ventral fins. It has a slightly negative slope that indicates directional instability. The -0.0002 value is used later in the parametric analysis of  $C_{n\beta}$ .

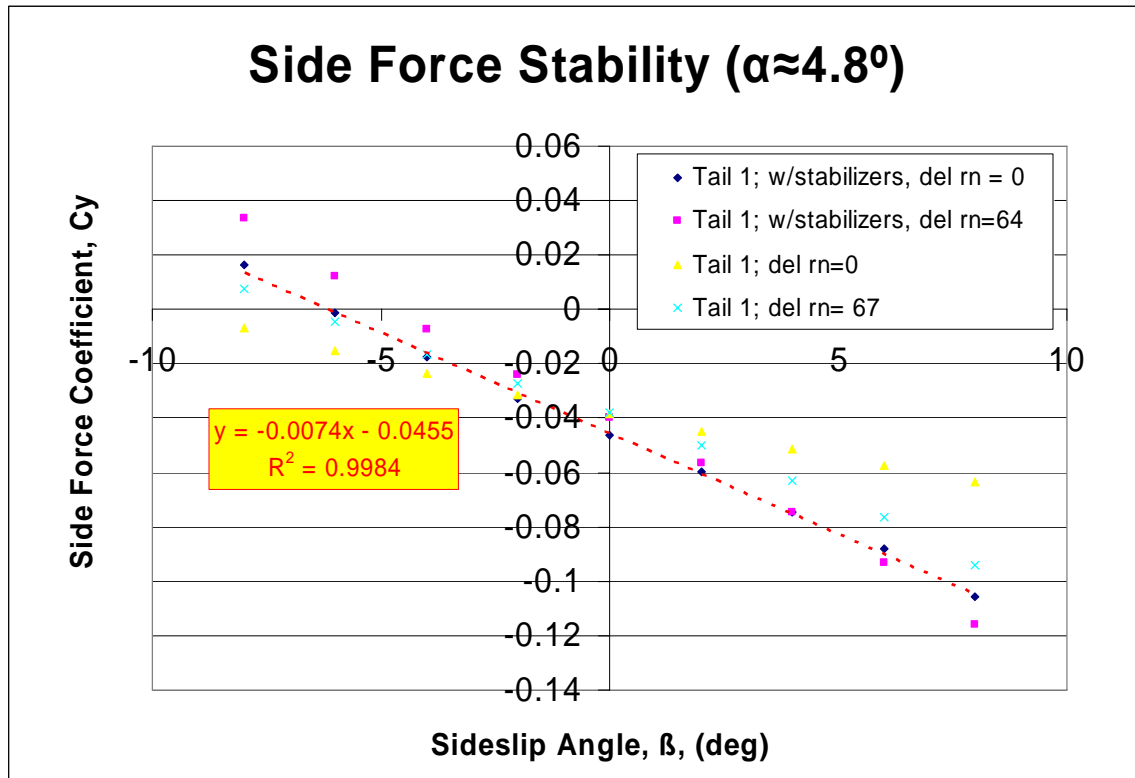


Figure 12.  $C_{Y\beta}$  for Zero Tail Inputs (Leveron, 2005)

Note the  $\beta$ -dependent fits do not pass through the origin of the plot as theory dictates.

Parga's thesis explains that the tail is mounted asymmetrically along the longitudinal axis due to actuating mechanism and fabrication limitations. Figure 13 illustrates and quantifies the slight asymmetry of the tail that helps account for the non-zero coefficients for the zero tail input and zero angle of sideslip condition.

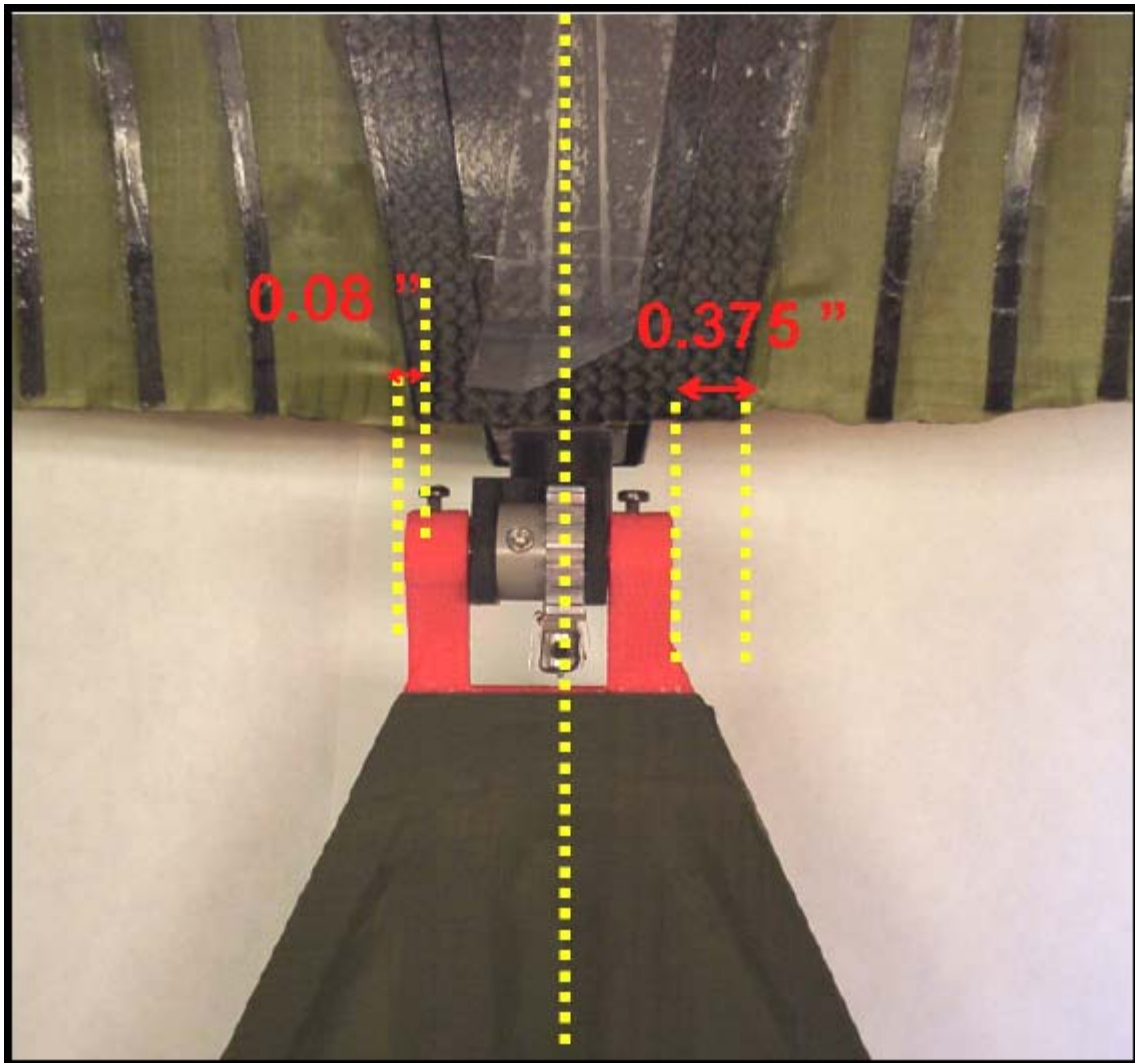


Figure 13. Tail Mounting Asymmetry (Parga, 2004:158)

This asymmetry could be corrected in production versions, and unnecessarily complicates the coupled non-linear system. Therefore, these biases are zeroed when they are combined into the overall coefficient calculations. The zero tail input math models form the first term of the overall coefficient equations, and are shown in Equation 14. The remaining contributors are the tail effect and any damping term that may apply.

$$\begin{aligned}
C_D(\alpha) &= 0.00003\alpha^3 + 0.0008\alpha^2 + 0.0052\alpha + 0.078 \\
C_L(\alpha) &= -0.0002\alpha^3 - 0.0014\alpha^2 + 0.0924\alpha + 0.7948 \\
C_m(\alpha) &= 0.000009\alpha^4 - 0.00007\alpha^3 - 0.0018\alpha^2 - 0.011\alpha + 0.0087 \\
C_\ell(\beta) &= -0.0018\beta \\
C_n(\beta) &= 0.0005\beta \\
C_Y(\beta) &= -0.0074\beta
\end{aligned} \tag{14}$$

The tail control effectiveness curve fits include the  $\alpha$  and  $\beta$  fits that correspond to the coefficient of interest. Figures 14 through 19 illustrate the raw data and the curve fit equation for each of the coefficients shown in the same order as the  $\alpha$  and  $\beta$  curve fits. Each comparison uses a contour plot above and a corresponding surface plot below with the same z-axis and color bar scale for a full visualization of the function. The red equation listed between the contour and surface plots is the second order quadratic interaction model with the  $x_1$  to  $x_9$  fit terms. Note that the plots' negative  $\delta_e$  values correspond to a tail up deflection and a positive  $\delta_m$  corresponds to a clockwise rotation of the tail from the aft perspective.



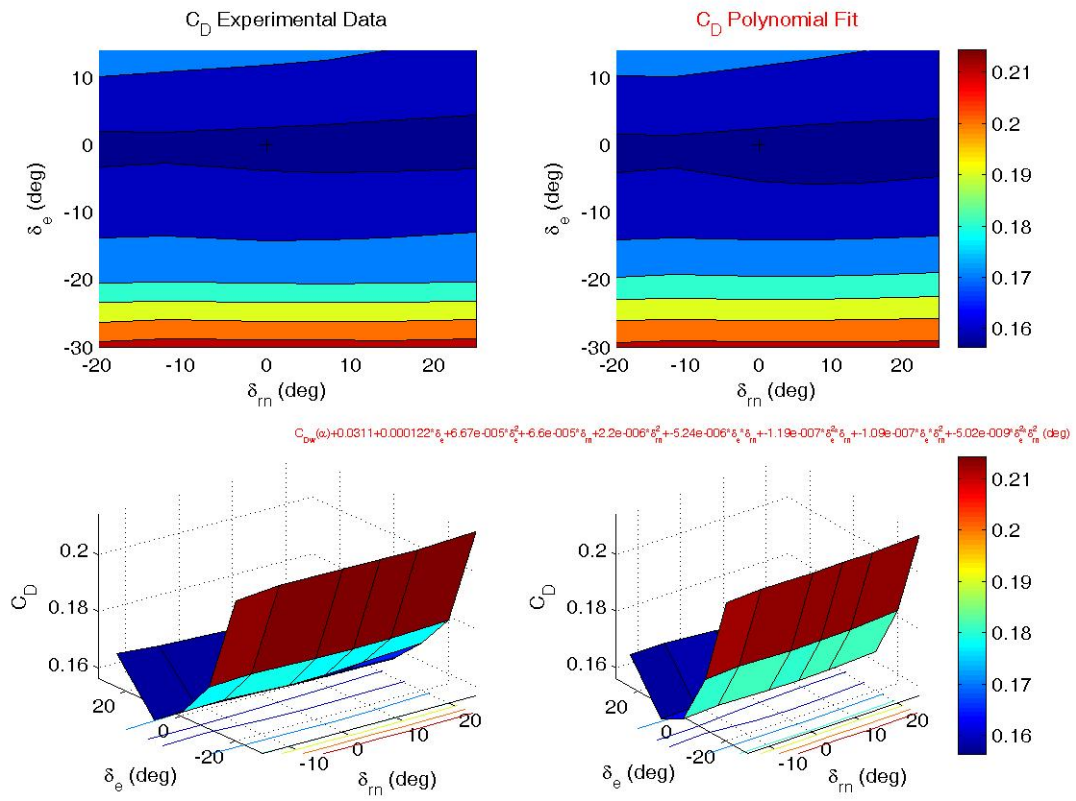


Figure 14. Tail Effect on Drag Coefficient

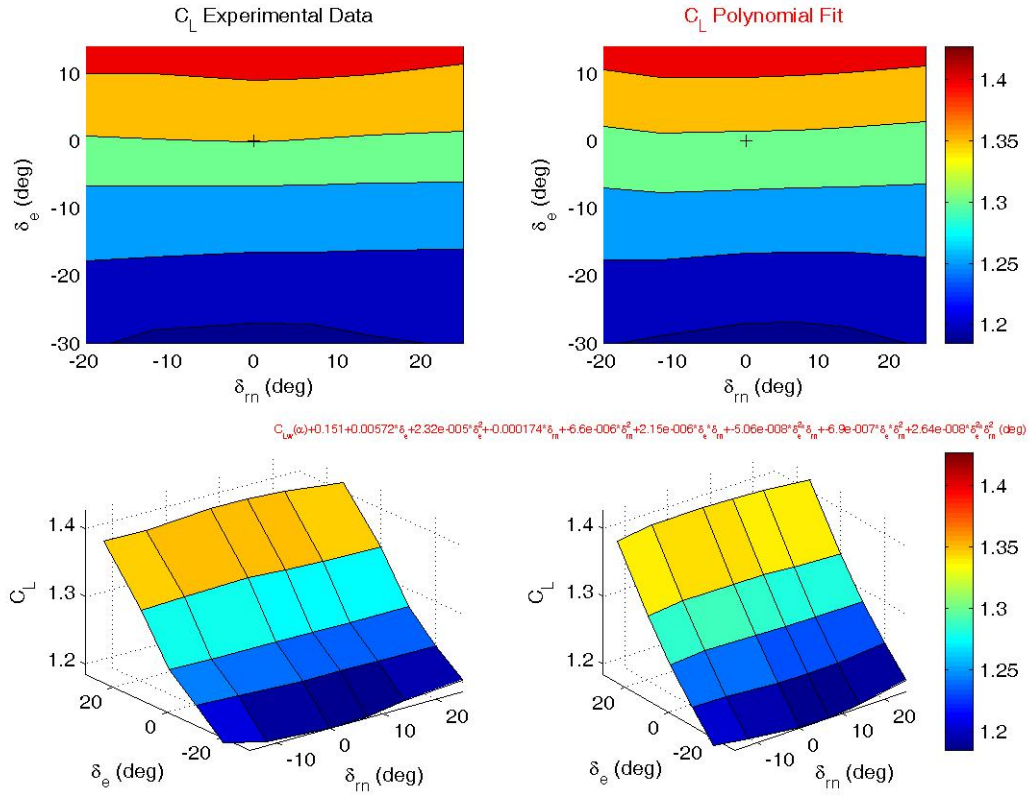


Figure 15. Tail Effect on Lift Coefficient

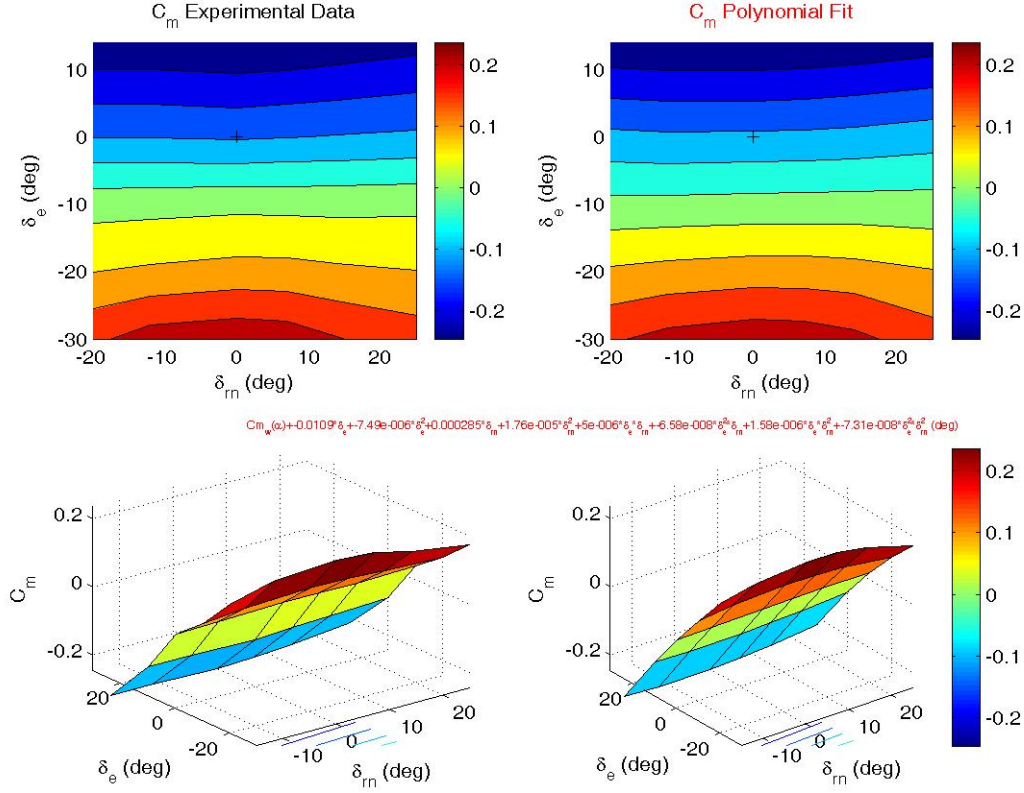


Figure 16. Tail Effect on Pitching Moment Coefficient

The  $\alpha$ -dependent coefficients of Figures 14 through 16 get their control effectiveness primarily from the  $\delta_e$  input. A slight dependence on the  $\delta_m$  inputs is evident, and would probably exaggerate the curvature of the function if the range continued out toward  $\pm 60^\circ$  of tail rotation.

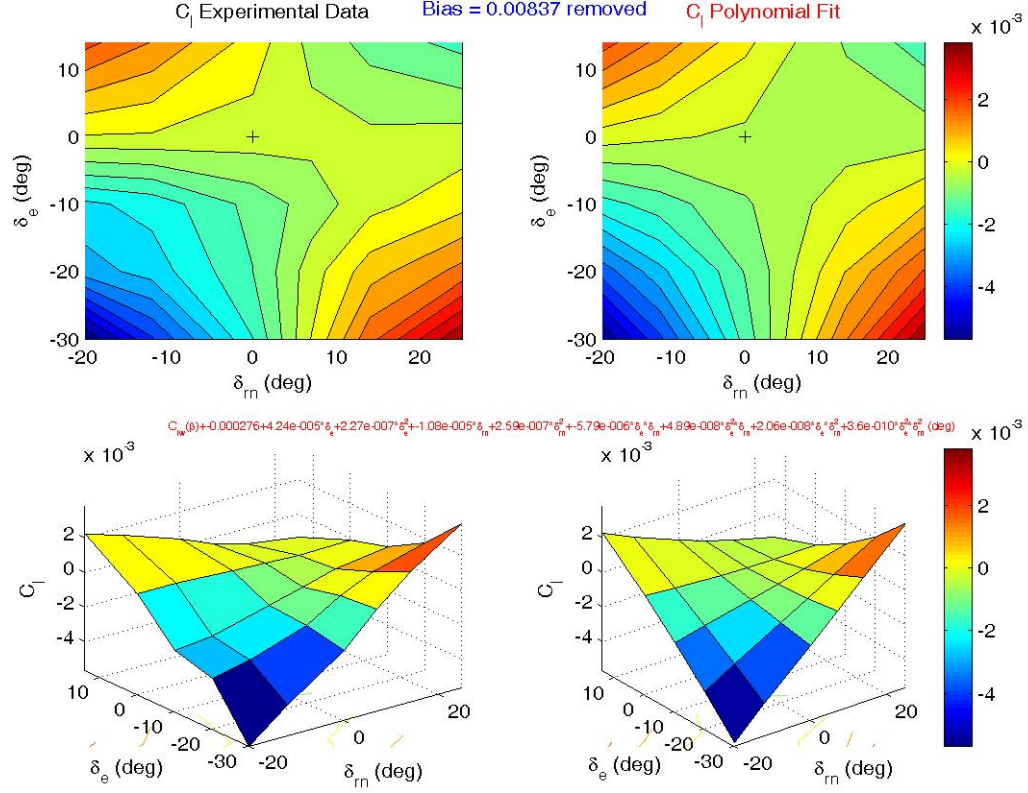


Figure 17. Tail Effect on Roll Moment Coefficient

The previously discussed tail mounting asymmetries cause counter-theoretical non-zero values at the  $\delta_e=0$ ,  $\delta_m=0$ , and  $\beta=0$  degrees point in this data set as well. The  $C_l(\beta)$ ,  $C_n(\beta)$ , and  $C_Y(\beta)$  coefficients of Figures 17 through 19 have the measurement bias removed from the raw data before fitting. This makes the saddle functions centered about zero and improves tail control authority for trimming and maintaining SLUF.

Even with the biases removed, there is a distinct asymmetry in the  $C_l$  data and curve fit. This asymmetry could lead to ill behaved roll stability. Ideally, the contour lines should be symmetric about the  $\delta_m=0$  axis. This anomaly is addressed later in an investigation of data smoothing for improved simulation.

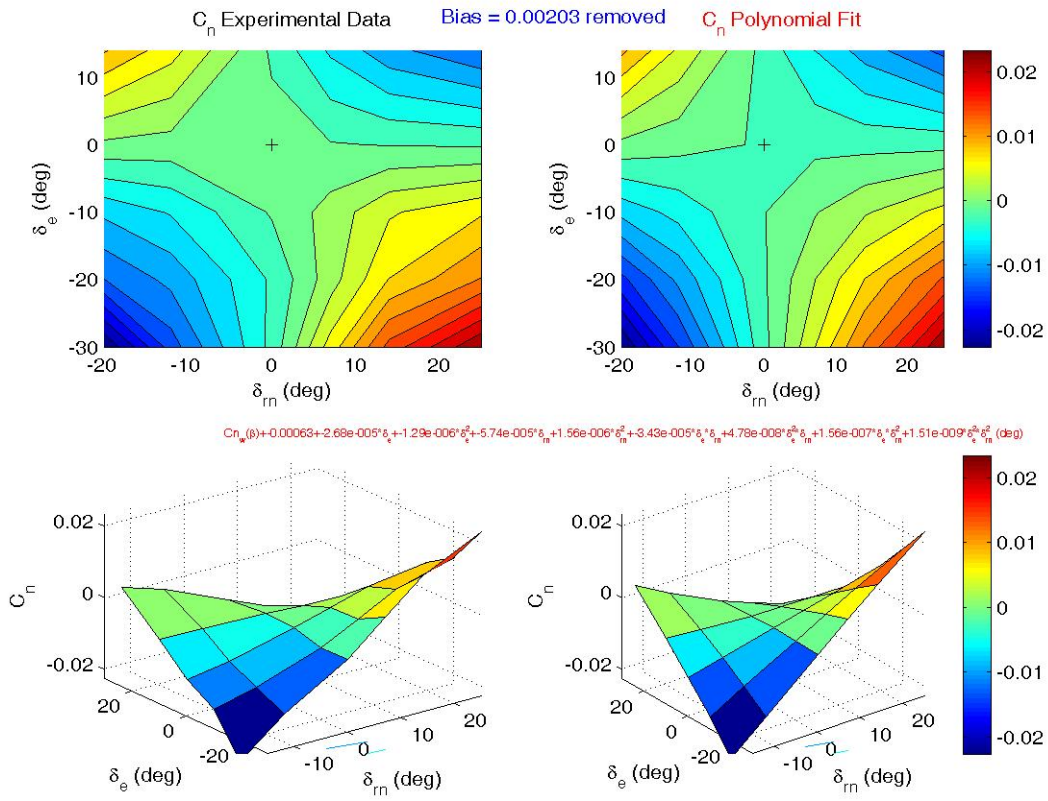


Figure 18. Tail Effect on Yaw Moment Coefficient

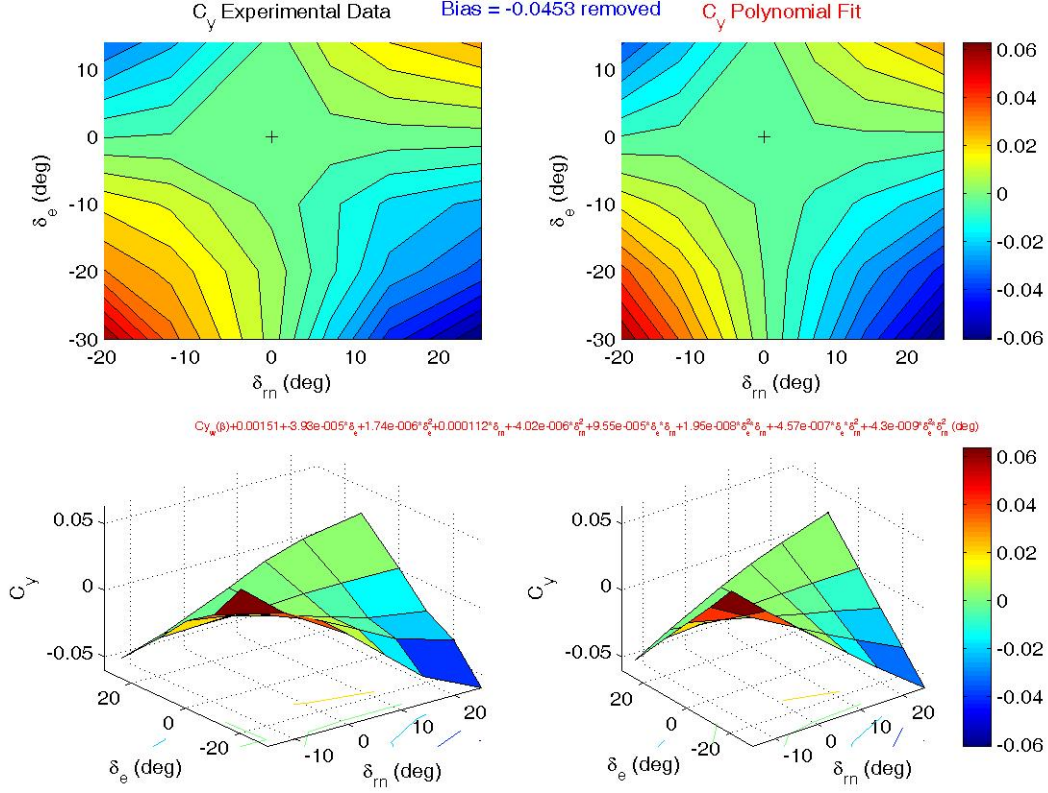


Figure 19. Tail Effect on Side Force Coefficient

Figures 18 and 19 show well behaved lines of symmetry from the ridges and troughs of the saddles through the origin of the plot. A summary of the tail effectiveness curve fit math models is listed in Equation 15. (15)

$$\begin{aligned}
 \Delta C_D(\delta_m, \delta_e) &= 0.0311 + 0.000122\delta_m + 6.67E^{-5}\delta_m^2 - 6.60E^{-5}\delta_e + 2.20E^{-6}\delta_e^2 - 5.24E^{-6}\delta_m\delta_e - 1.19E^{-7}\delta_m^2\delta_e - 1.09E^{-7}\delta_m\delta_e^2 - 5.02E^{-9}\delta_m^2\delta_e^2 \\
 \Delta C_L(\delta_m, \delta_e) &= 0.151 + 0.00572\delta_m + 2.32E^{-5}\delta_m^2 - 0.000174\delta_e - 6.60E^{-6}\delta_e^2 + 2.15E^{-6}\delta_m\delta_e - 5.06E^{-8}\delta_m^2\delta_e - 6.90E^{-7}\delta_m\delta_e^2 + 2.64E^{-8}\delta_m^2\delta_e^2 \\
 \Delta C_m(\delta_m, \delta_e) &= -0.257 - 0.0109\delta_m - 7.49E^{-6}\delta_m^2 + 0.000285\delta_e + 1.764E^{-5}\delta_e^2 + 5.00E^{-6}\delta_m\delta_e - 6.58E^{-8}\delta_m^2\delta_e + 1.58E^{-6}\delta_m\delta_e^2 - 7.31E^{-8}\delta_m^2\delta_e^2 \\
 \Delta C_l(\delta_m, \delta_e) &= -0.000276 + 4.24E^{-5}\delta_m + 2.27E^{-7}\delta_m^2 - 1.08E^{-5}\delta_e + 2.59E^{-7}\delta_e^2 - 5.79E^{-6}\delta_m\delta_e + 4.89E^{-8}\delta_m^2\delta_e + 2.06E^{-8}\delta_m\delta_e^2 + 3.61E^{-10}\delta_m^2\delta_e^2 \\
 \Delta C_n(\delta_m, \delta_e) &= -0.000431 + 2.98E^{-5}\delta_m - 4.9272E^{-7}\delta_m^2 - 5.64E^{-5}\delta_e + 1.08E^{-6}\delta_e^2 - 3.43E^{-5}\delta_m\delta_e + 5.21E^{-8}\delta_m^2\delta_e + 1.63E^{-7}\delta_m\delta_e^2 - 3.99E^{-10}\delta_m^2\delta_e^2
 \end{aligned}$$



$$\Delta C_l(\delta_m, \delta_e) = 0.0010858 + 1.10E^{-5}\delta_m + 2.56E^{-7}\delta_m^2 + 0.000110\delta_e - 3.00E^{-6}\delta_e^2 + 9.57E^{-5}\delta_m\delta_e + 1.17E^{-8}\delta_m^2\delta_e - 5.77E^{-7}\delta_m\delta_e^2 - 7.65E^{-10}\delta_m^2\delta_e^2$$

The combined curve fit results of Equations 14 and 15 led to anticipation of a near-zero sum of the forces and moments for a single point in time, and potentially divergent lateral-directional stability during the open loop simulation.

### Open Loop Simulation

The next step towards closed loop control of an aircraft is successful modeling of its open loop behavior. All of the math models developed from the curve fits are incorporated into the *ForcesMoments.m* file found in Appendix C. The resulting forces and moments are then used by the *Trimmer.m* file (Appendix D) to perform a static optimization. The effect of trim speed,  $V_T$ , on  $\alpha$  and  $\delta_e$  is seen in Figure 20.

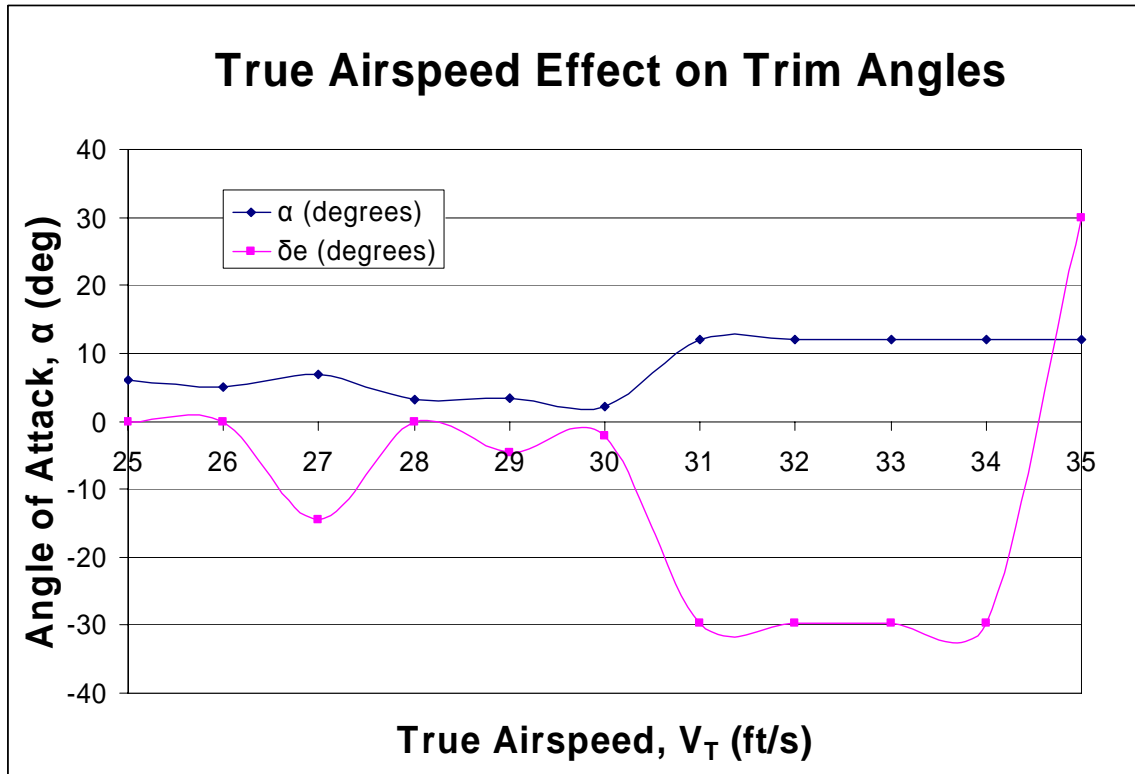


Figure 20. True Airspeed Effect on Trim Angles

This is an interesting result because of the sharp tail deflection to the full up position and high angle of attack at trim speeds beyond the 30 ft/s. This indicates a  $Cm_\alpha$  sensitivity that is probably due to  $cg$  placement when the data were taken. There is a somewhat narrow band between 28-30 ft/s (approximately 20 mph) where both  $\alpha$  and  $\delta_e$  are small and nearly equal. Ultimately, 30 ft/s is chosen due to mission requirements and simplicity and yields the following trimmed values for the unbiased data with no artificial stability or damping augmentation:

$$u = \begin{bmatrix} T \\ \delta_e \\ \delta_m \end{bmatrix} = \begin{bmatrix} 0.0862 \text{ lb}_f \\ -2.07^\circ \\ 0^\circ \end{bmatrix} \quad x = \begin{bmatrix} V_T \\ \gamma \\ \alpha \\ q \\ p \\ \mu \\ \beta \\ r \\ \chi \\ \xi \\ \eta \\ h \end{bmatrix} = \begin{bmatrix} 30 \text{ ft} / \text{s} \\ 0 \\ 2.14^\circ \\ 0 \\ 0 \\ 0 \\ 0 \\ -3.18E^{-32} \\ -2.65E^{-31} \\ -3.40E^{-32} \\ 6.76E^{-34} \\ 50 \text{ ft} \end{bmatrix}$$

$$\begin{aligned} L &= 0.781 \text{ lb}_f \\ D &= 0.0864 \text{ lb}_f \\ Y &= 0.00111 \text{ lb}_f \\ m &= -0.000220 \text{ ft} \cdot \text{lb}_f \\ n &= -0.000806 \text{ ft} \cdot \text{lb}_f \\ \ell &= -0.000505 \text{ ft} \cdot \text{lb}_f \end{aligned}$$

Of note within this output is  $\delta_e \approx \alpha$ , flight path,  $\beta$ , and all rates about the aircraft are hard zero or machine zero. Also, the trimmed forces and moments balance to ten-thousandths of a pound. Unfortunately, the open loop performance of this system yields the divergent flight path displayed in Figure 21 and states in Figure 22.



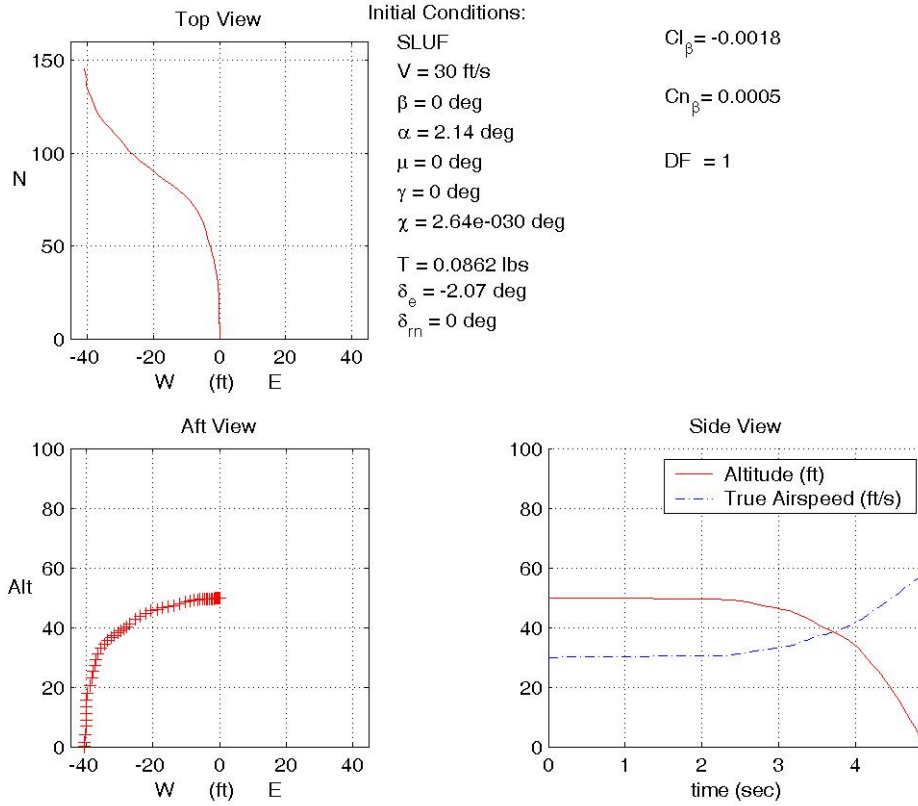


Figure 21. Flight Path Results of Unbiased Data Unaugmented MAV

This simulation is based on the MAV aircraft data with aft ventral fins where  $Cl_\beta = -0.0018$ ,  $Cn_\beta = 0.0005$ , and the roll damping is based on Nelson's rigid aircraft estimating equation. The DF term in the upper right of Figure 21 is an artificial Damping Factor multiplier applied later to the  $C_{lp}$  term for parametric analysis.

The aircraft departs the trim heading after approximately 25 ft (less than one second at 30 ft/s) and begins losing altitude after 1.5 seconds. Impact occurs at 4.90 seconds. It is not clear from this plot which mode drives the simulation to depart SLUF. Further inspection of the individual states identifies the crucial states dominating the undesired dynamic behavior. Figure 22 lists all 12 of the states in strip chart fashion.

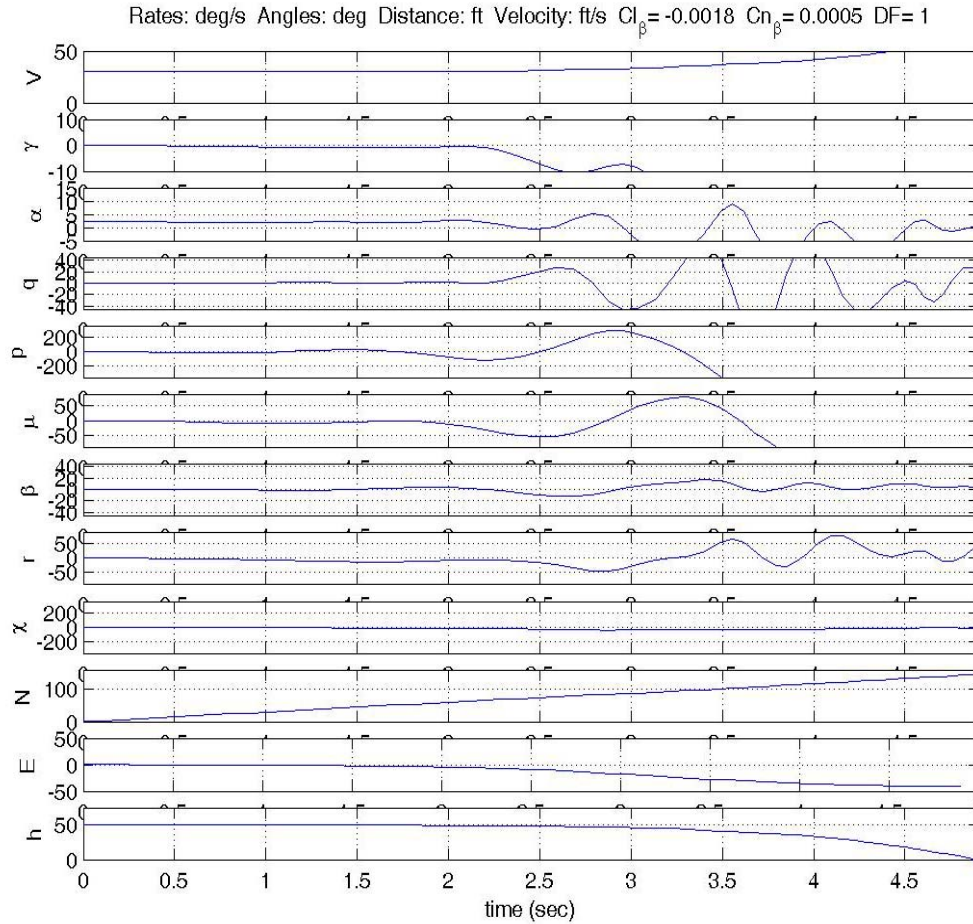


Figure 22. Strip Chart Results of Unbiased Data Unaugmented MAV

A review of all 12 states quickly keys into the fifth and sixth strip charts for roll rate,  $q$ , and bank angle,  $\mu$  as the first and second departures from trim, respectively. The North position may look as though it is a bad actor, but it is simply increasing with time as the aircraft tracks down range as expected. The sideslip angle,  $\beta$ , and yaw rate,  $r$ , are also of interest since the aircraft seems to depart lateral-directionally prior to any longitudinal problems. Figure 23 zooms in on the crucial drivers for more detailed analysis.

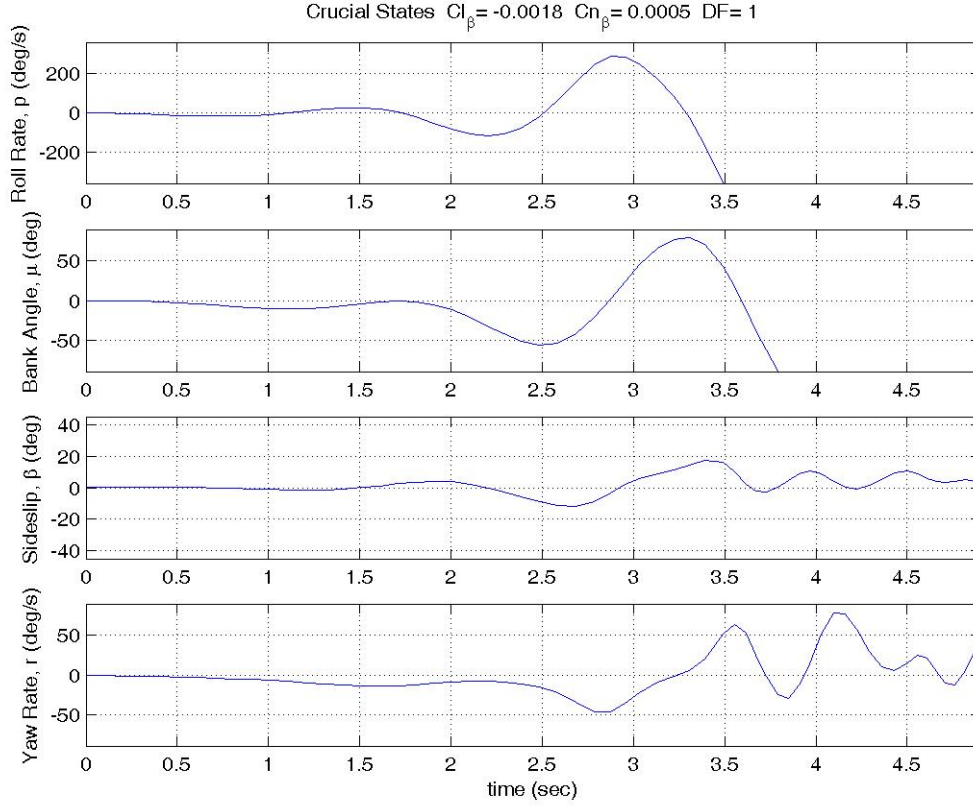


Figure 23. Crucial Strip Results of Unbiased Data Unaugmented MAV

Notice the scale for  $p$  is  $\pm 360$  degrees per second. It seems to begin divergence by 0.75 seconds and after only two oscillations, it is rapidly exceeding one full rotation per second in roll. Logically, the bank angle follows closely behind the roll rate behavior. Although the  $\mu_{dot}$  ODE is the most involved of the 12 EOMs, small angle approximation simplifies it to equal  $p$ . That means  $\mu$  will essentially be one time step behind  $p$  in its behavior. Once the aircraft diverges in roll, the other states follow.

The poor stability requires more in-depth analysis. The linearization process of Chapters II and III is applied to find eigenvalues of the partitioned system. The

linearized state space  $A$  matrix and corresponding eigenvalues for the unbiased basic open loop MAV is listed in Table 3.

		A Matrix									
ph	V	-0.23661922	-32.17	-15.177572	0	0	0	0.0457932	0		
	$\gamma$	0.071202519	0	4.6096779	0	0	-0.001526	0	0		
sp	$\alpha$	-0.071202519	0	-4.6096779	1	0	0	0	0		
	q	-0.000177028	0	-41.593731	0	0	0	0	0		
roll	p	-0.017977396	0	0	0	-0.172605	0	-74.17685	0		
	$\mu$	0	0.0015264	2.5863E-32	0	0.9992254	0	0.0006998	0.0393537		
dr	$\beta$	5.08813E-05	0	0	0	0.0393537	1.0723333	-0.399951	-0.999225		
	r	-0.007575485	0	0	0	0	0	5.6770585	0		
Eigenvalues:		-0.11831 ± 1.5088i	-2.3048 ± 6.0234i	0		-0.19998 ± 2.3733i					
Entire A Matrix		1.627 ± 4.2801i				-0.1726					
		-3.8273									
		-2.4595 ± 5.8976i									
		0.00030967									
		0.020129 ± 1.5031i									

The red fonts in the eigenvalue blocks identify non-negative eigenvalues that are not stable. The roll mode has a zero eigenvalue representing the free integrator of roll rate to bank angle. The surprising result and source of the instability is the coupled lateral-directional mode inside the dashed box that has two positive eigenvalues—one large complex value and one small real value that indicate instability. This is odd because each separate mode has stable eigenvalues that support the traditional force and moment coefficient slope indicators of stability. However, when the separately stable portions of the lateral-directional mode are coupled, they are unstable.

### Artificial Stability Improvements

Since the static roll stability derivatives are clearly stable, the only remaining source for dynamic divergence within the lateral-directional terms. A parametric sensitivity study was done by varying yaw stability, roll stability (otherwise known as

dihedral effect), and roll damping. Roll moment damping due to roll rate,  $C_{lp}$ , is the only term added to the  $C_l$  coefficient equation as described in Chapters II and III. Originally, the roll damping due to vertical tail contributions was neglected because there is no vertical tail on the primary design of the rotatable tail MAV. However, Leveron tested a configuration with aft ventral fins, and that data is used in the simulation. The aft ventral fins combined with the notion of improved damping from flexible wings as opposed to rigid wings leads to the question of open loop stability with graduated increases in roll damping via an artificial Damping Factor (DF). Another stability parameter easily improved by simple design modifications is  $Cn_\beta$ . Leveron successfully changed the static yaw stability from slightly unstable (-0.0002) to slightly stable (+0.0005) by adding small ventral fins. Bigger ventral fins could further improve the  $Cn_\beta$  term. Additionally, the  $Cl_\beta$  for dihedral effect seemed high based on Leveron's analysis, so its value was decreased all the way to zero to study its affect (Leveron, 2005:67). Table 4 lists the parameter variations and the corresponding times of flight and linearized eigenvalues as calculated by Matlab's *linmod.m* about the trimmed initial condition.

Table 4. Parametric Lateral-Directional Sensitivity Analysis

Asymmetric control maps:

Unbiased, NO data mirroring

Values represent time (sec) from  $h_o = 50$  ft to  $h_f = 0$  ft

$Cn_\beta$	$C_{lp}$	$10*C_{lp}$	$50*C_{lp}$	$100*C_{lp}$
-0.0002	4.15	5.43	7.51	5.35
0.0005	4.90	6.16	19.60	27.13
0.0018	6.04	8.20	28.78	41.00
0.0030	7.23	10.85	32.67	46.28
0.0100	6.57	15.91	39.48	54.92

Right Most Poles for Longitudinal Modes (phugoid)

$Cn_\beta$	$C_{lp}$	$10*C_{lp}$	$50*C_{lp}$	$100*C_{lp}$
-0.0002	-0.11723	-0.11723	-0.11723	-0.11723
0.0005	-0.11723	-0.11723	-0.11723	-0.11723
0.0018	-0.11723	-0.11723	-0.11723	-0.11723
0.0030	-0.11723	-0.11723	-0.11723	-0.11723
0.0100	-0.11723	-0.11723	-0.11723	-0.11723

$Cl_\beta = -0.0018$  Right Most Poles (Lateral-Directional Modes)

$Cn_\beta$	$C_{lp}$	$10*C_{lp}$	$50*C_{lp}$	$100*C_{lp}$
-0.0002	1.93740	1.41800	0.14058	0.02010
0.0005	1.62690	1.18440	0.10045	0.02197
0.0018	1.10870	0.82457	0.05452	0.03257
0.0030	0.76154	0.58782	0.03142	0.03518
0.0100	0.11937	0.090166	0.036948	0.038408

$Cn_\beta$	$Cl_\beta$	$0.5*Cl_\beta$	$0.25*Cl_\beta$	$0*Cl_\beta$
-0.0002	1.93740	1.55860	1.26270	1.29370
0.0005	1.62690	1.16750	0.77369	0.03806
0.0018	1.10870	0.59786	0.23426	0.03819
0.0030	0.76154	0.32336	0.06303	0.03821
0.0100	0.11937	0.01999	0.01998	0.03823

$C_{lp}$  is not scaled by any DF in the bottom sub-matrix

Multiple points are evidenced in Table 4. First, the time of flight to ground impact does not directly correlate to the stability indicated by the eigenvalue. Second, the longitudinal modes (the phugoid is the right most pole) are completely unaffected by any of the variable changes. This uncoupled behavior is advantageous for future control development. Thirdly, the changes in yaw stability, roll stability, and roll damping all show significant improvement in the lateral-directional eigenvalues, but none of them individually or collectively create a stable (negative) eigenvalue for the coupled mode. The final point from Table 4 is that decreasing  $Cl_\beta$ , rather than increasing it like the other stability derivatives, significantly improves linearized lateral-directional eigenvalue stability indicators all the way to  $Cl_\beta = 0$ . The  $Cl_\beta$  were reduced in conjunction with the increases in  $Cn_\beta$  while  $Cl_p$  was held constant at its original estimate value. This combination of improved yaw stability and reduced dihedral effect yielded more improvement than the combination of yaw stability improvement with unachievably high DF scaling values on the  $Cl_p$  term.

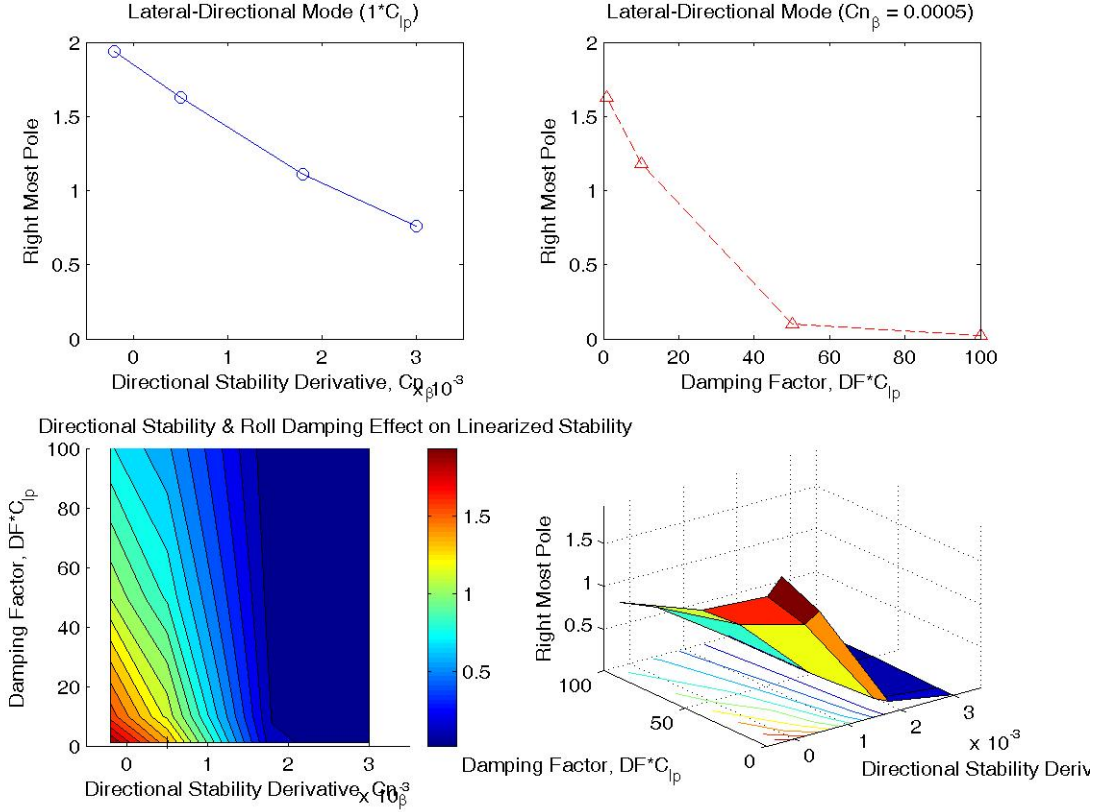


Figure 24. Linearized Eigenvalue Stability Improvement as a Function of  $Cn_\beta$  and  $C_{l_p}$

Not surprisingly, as the damping and directional stability increase, the aircraft is more stable, flies longer before hitting the ground, and has more stable (less positive) linearized eigenvalues. A clear change in performance occurs between the  $DF=10$  and  $DF=50$ . The aircraft goes from being dynamically divergent in roll to being dynamically stable in roll. Once the roll oscillations are damped, the directional stability begins to dominate going from directional divergence to spiral stability as  $Cn_\beta$  increases. An example of flight path for increase yaw stability and damping conditions is shown in Figure 25.



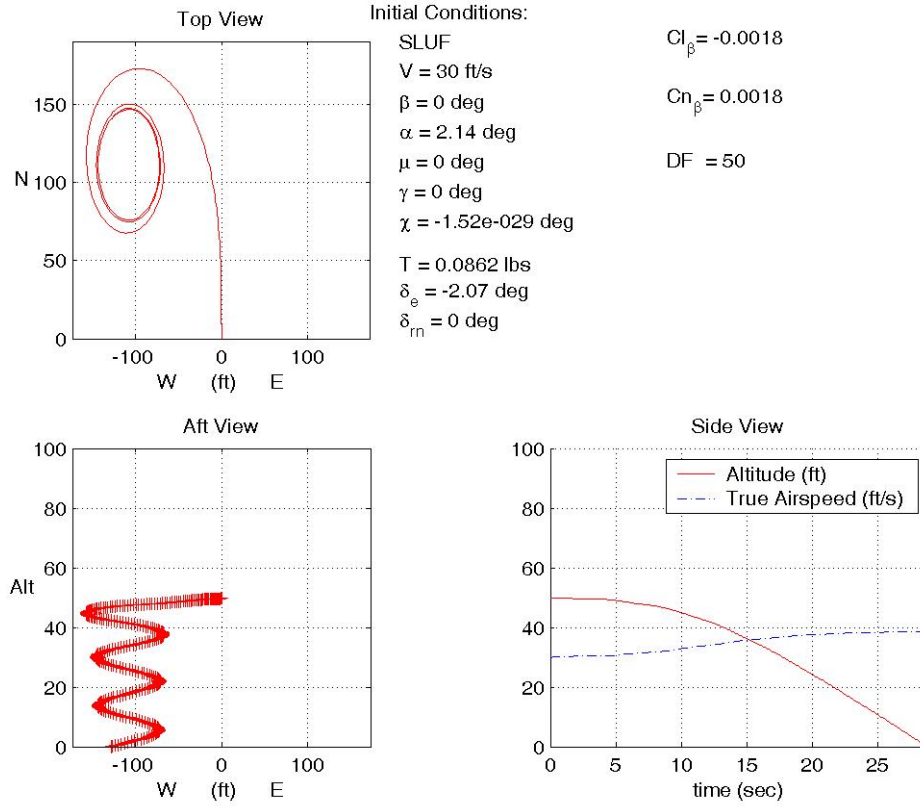


Figure 25. Flight Path for  $DF=50$  and  $Cn_\beta = +0.0018$

Unfortunately, the slightly non-zero trimmed forces and moments—especially the side force and yaw moments—integrate over time to perturb the aircraft from SLUF. The aircraft exhibits a stable spiral roll mode due to the perturbations, but does not achieve true SLUF. Further review of the strips charts reveals more insight about the simulation. Figure 26 shows all 12 states from the augmented open loop stability.

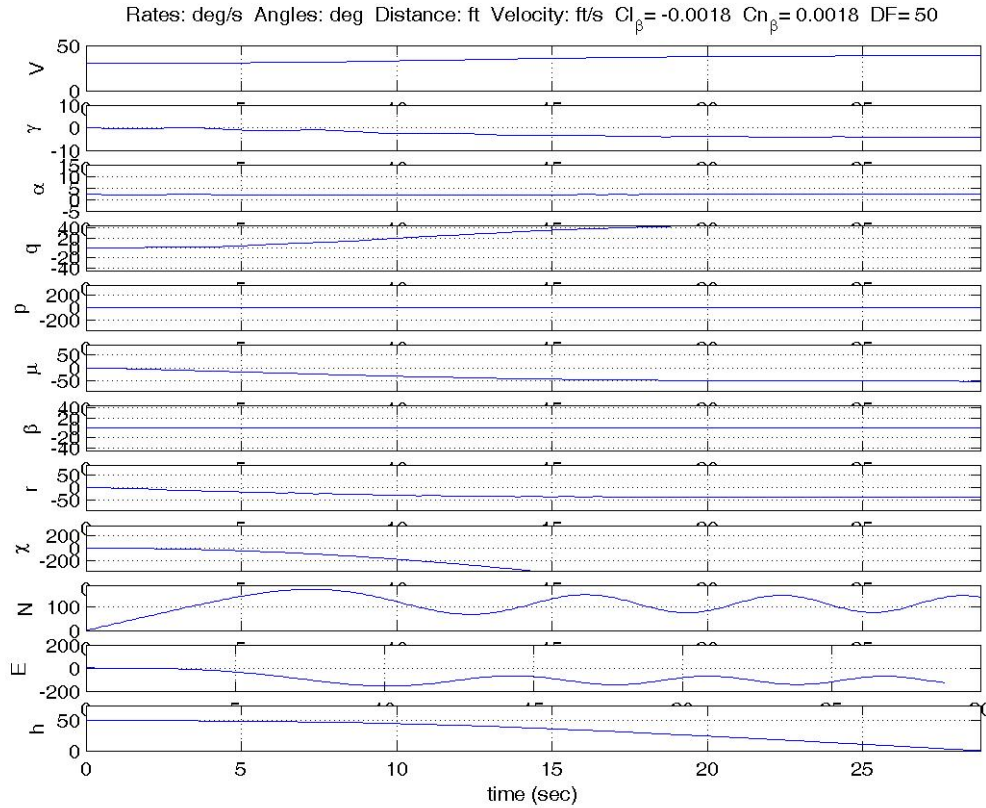


Figure 26. Strip Charts for DF=50 and  $Cn_\beta = +0.0018$

Despite near-zero damped roll rate and sideslip angle, the aircraft begins to develop negative (left) bank angle and yaw rate within approximately 1.5 seconds of flight. As bank angle and yaw rate increase in magnitude, the pitch rate increases and the flight path angle decreases while the aircraft proceeds into a constant thrust spiral dive. Magnified scales on the strip charts of the crucial states helps clarify the deviation from SLUF. Figure 27 zooms in closely about the states of interest and reveals small magnitude, damped roll and yaw rates.

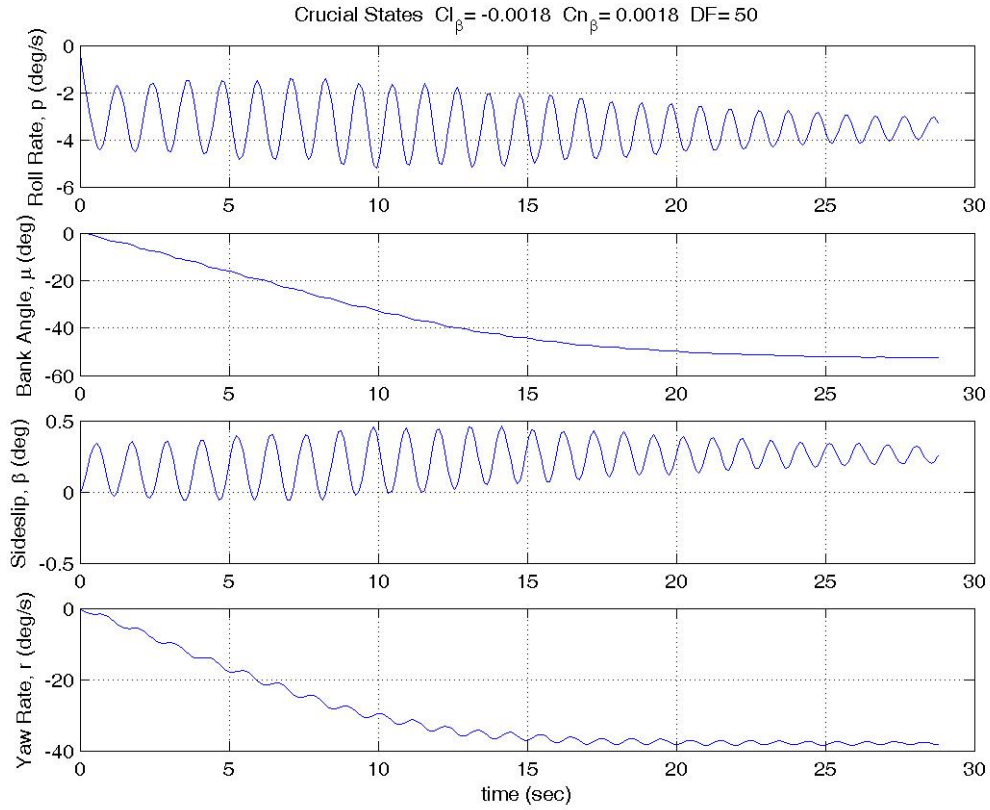


Figure 27. Crucial States for  $DF=50$  and  $Cn_{\beta} = +0.0018$

The roll rate lightly damps to a steady state value approximately equal to -3 deg/s that integrates bank angle immediately towards a steady state value of approximately 50 deg left bank. This residual steady state roll rate must be due to some unaccounted bias or asymmetry in the roll moment coefficient.

Recall the asymmetry of the  $C_l$  tail effectiveness plots in Figure 17. Envision lines of symmetry along the ridge and trough of the saddle plot in the upper right hand corner of the figure. The intersection of those lines place the true zero point at approximately  $\delta_m=5^\circ$  and  $\delta_e=-5^\circ$ . Additionally, an ideal  $C_l$  fit would have a mirror image

about the  $\delta_m=0^\circ$  axis to coincide with a symmetric aircraft. The real-world experimental values illustrate the fabrication imperfections, etc. that result in a *bent* MAV.

A more idealistic MAV is required to achieve true open loop SLUF. The lateral-directional data ( $C_b$ ,  $C_n$ , and  $C_Y$ ) are all modified to reflect true zero coefficients at  $\delta_m=0^\circ$  rather than very small non-zero values. Furthermore, the  $C_l$  tail effectiveness data are modified by neglecting the  $\delta_m=-12^\circ$  and  $\delta_m=-20^\circ$  data sets. The bias at  $\delta_m=0^\circ$  and  $\delta_e=0^\circ$  of 0.008368709 is removed from all remaining measured values, and the  $\delta_m=+7^\circ$ ,  $+14^\circ$ , and  $+25^\circ$  are mirrored and sign changed prior to curve fitting. The modified data are available in Appendix I. Note that only the  $C_l$  data are properly mirrored. The  $C_n$  and  $C_Y$  values for  $\delta_m=0$  measurements are all equal to the value at  $\delta_m=0$  and  $\delta_e=0$  and will be zeroed by the curve fit files. All other coefficient values in the modified data file are disregarded. Figures 28 through 30 are the modified tail effectiveness curve fits resulting from the data modifications discussed above.

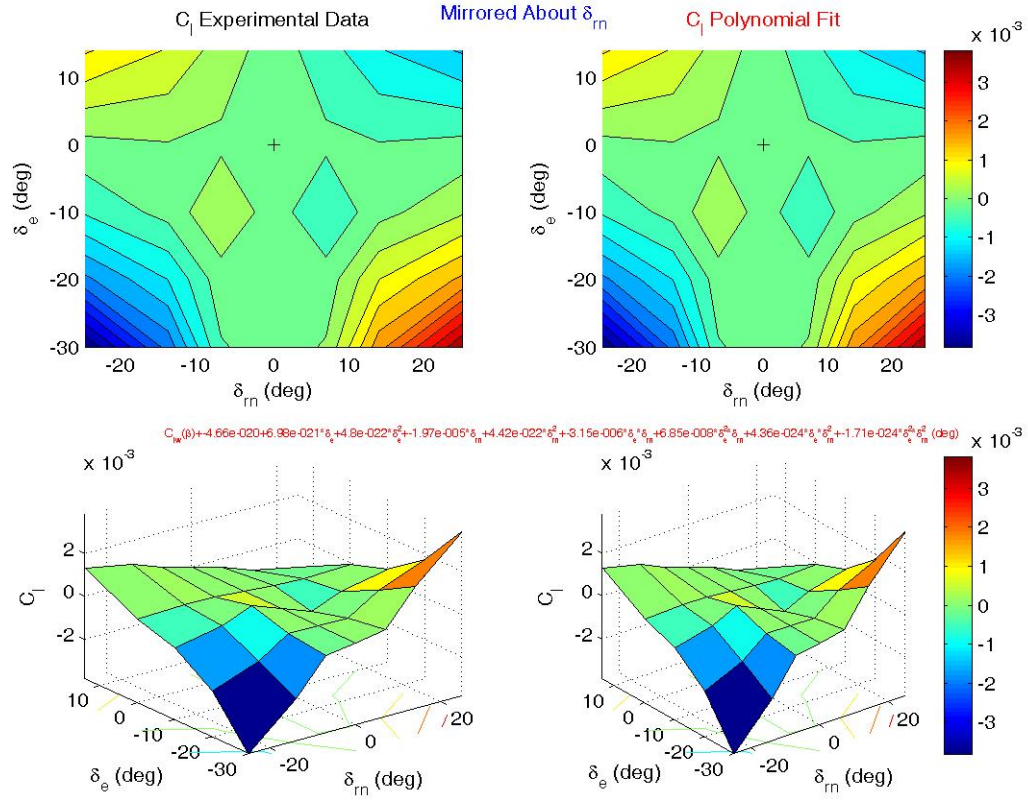


Figure 28. Modified  $C_l$  Curve Fit for  $\delta_m=0^\circ$  Bias Removal & Mirroring

The data modification achieves a mirrored symmetry about the  $\delta_m=0^\circ$  axis and true zero  $C_l$  values when  $\delta_m=0^\circ$ . However, an interesting pocket of sign reversal occurs centered at  $\delta_e=-10^\circ$  and  $\delta_m=\pm 7.5^\circ$ . This irregularity may be the result of wing downwash or fuselage interference in the relatively small tail input regime. Additional wind tunnel runs may be warranted to ensure that it was not an artifact of the data acquisition process.

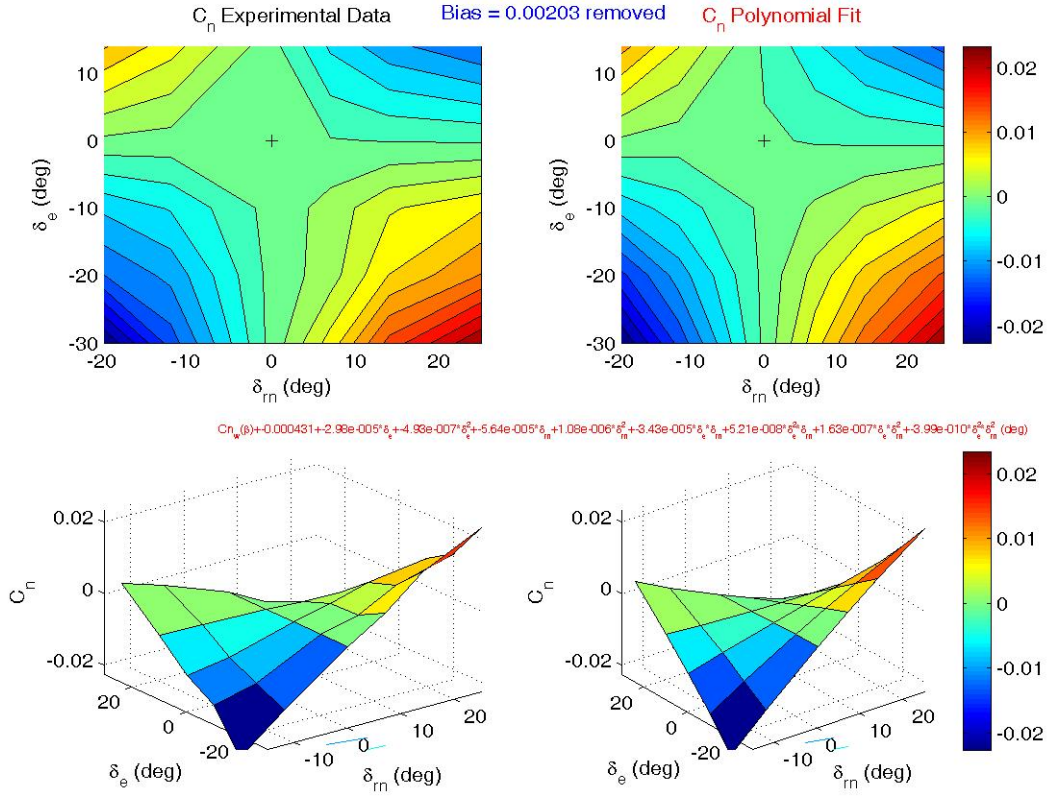


Figure 29. Modified  $C_n$  Curve Fit for  $\delta_m=0^\circ$  Bias Removal

The goal of bias removal is evidenced by the tightening of the contour lines to the  $\delta_m=0^\circ$  axis. This creates a small area of decoupling between pitch and lateral-directional control. The MAV is more easily trimmed for SLUF because a tail up deflection at  $\delta_m=0^\circ$  does not instantly induce a roll moment, yaw moment, or side force.

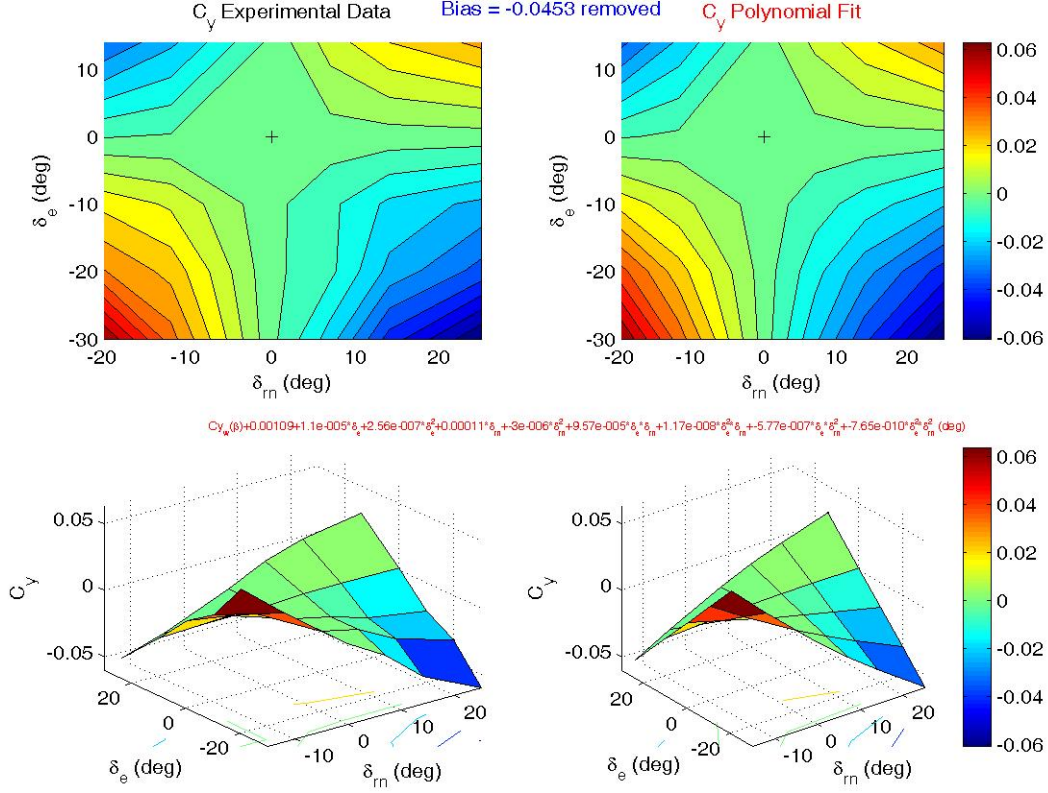


Figure 30. Modified  $C_Y$  Curve Fit for  $\delta_m=0^\circ$  Bias Removal

There is no noticeable difference in contour shape or magnitude of  $C_Y$  after the  $\delta_m=0^\circ$  bias removal. However, the  $x_1$  to  $x_9$  fit terms change slightly verifying the correction is applied. Further efforts to remove bias included dropping the first term (constant) of the tail effectiveness second order polynomial interaction fit for  $C_n$  and  $C_Y$  models. This is done inside the *ForcesMoments.m* file and does not affect the previously described curve fitting process. The resulting trim conditions and flight path of the open loop simulation is shown in Figure 31.



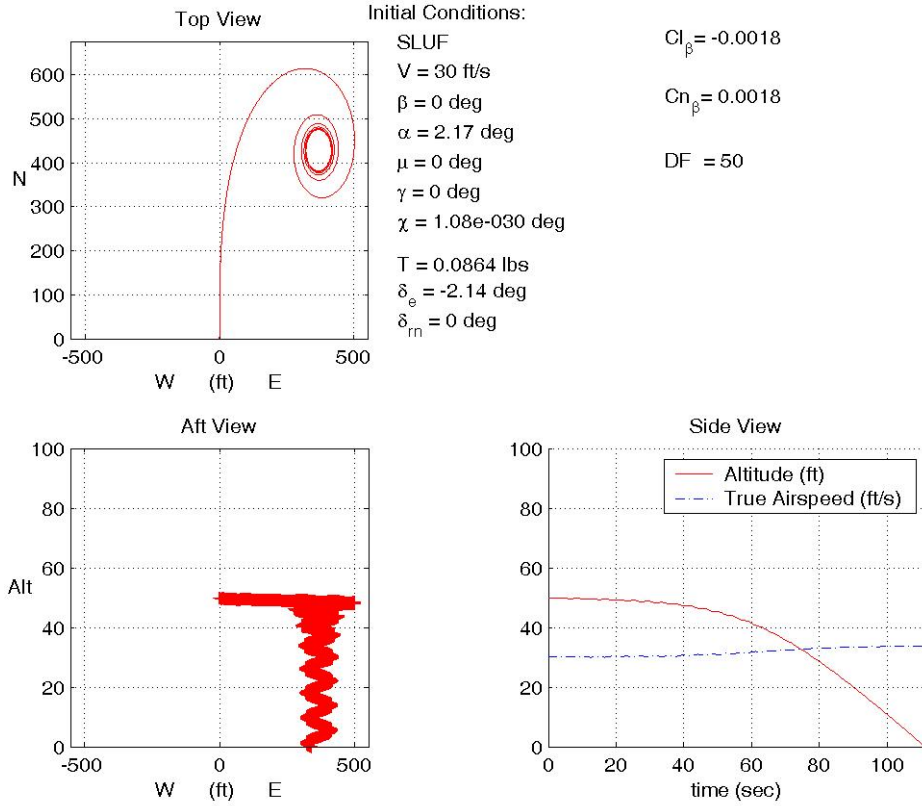


Figure 31. Flight Path for  $\delta_m=0^\circ$  Bias Removal & Mirroring

The effect of data mirroring and bias removal is dramatic. The overall time of flight improves from 28.78 seconds to 111.22 seconds. Additionally, the direction of deviation changes from left (W) to right (E). These significant changes are due to the reduced forces and moments at the initial trimmed condition. Below is a comparison of the trimmed forces and moments before (left) and after (right) the data modifications.

$$\begin{aligned} L &= 0.781 \text{ lb}_f \\ D &= 0.0864 \text{ lb}_f \\ Y &= 0.00111 \text{ lb}_f \\ m &= -0.000220 \text{ ft} \cdot \text{lb}_f \\ n &= -0.000806 \text{ ft} \cdot \text{lb}_f \\ \ell &= -0.000505 \text{ ft} \cdot \text{lb}_f \end{aligned}$$

$$\begin{aligned} L &= 0.781 \text{ lb}_f \\ D &= 0.0866 \text{ lb}_f \\ Y &= -1.55E^{-5} \text{ lb}_f \\ m &= -0.000215 \text{ ft} \cdot \text{lb}_f \\ n &= 8.58E^{-5} \text{ ft} \cdot \text{lb}_f \\ \ell &= -8.25E^{-20} \text{ ft} \cdot \text{lb}_f \end{aligned}$$



As expected, the most significant reductions in trimmed force and moment values were the roll moment and side force. The roll moment at initial trimmed conditions decreased by 16 orders of magnitude—reaching a practical machine zero. The side force decreased by two orders of magnitude with a sign change. The yaw moment also reduced by one order of magnitude and changed sign. Figures 32 and 33 illustrate the three dimensional flight path the the corresponding strip charts for the augmented stability derivatives and mirrored data set.

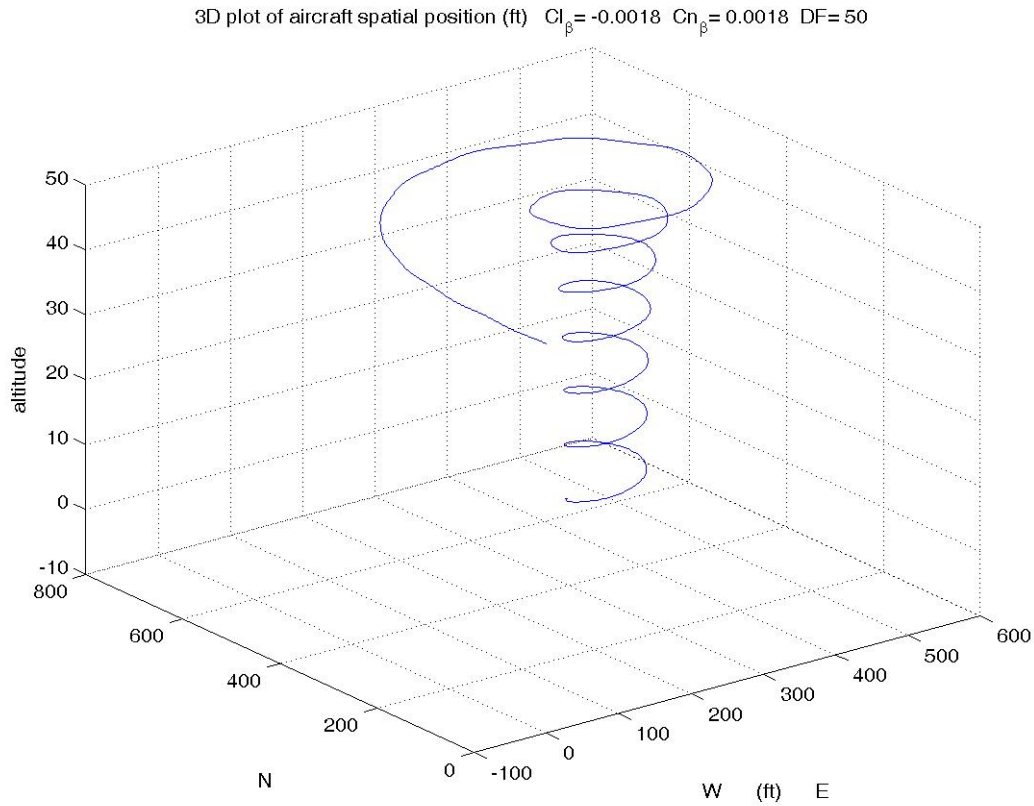


Figure 32. 3D Flight Path for  $\delta_m=0^\circ$  Bias Removal & Mirroring

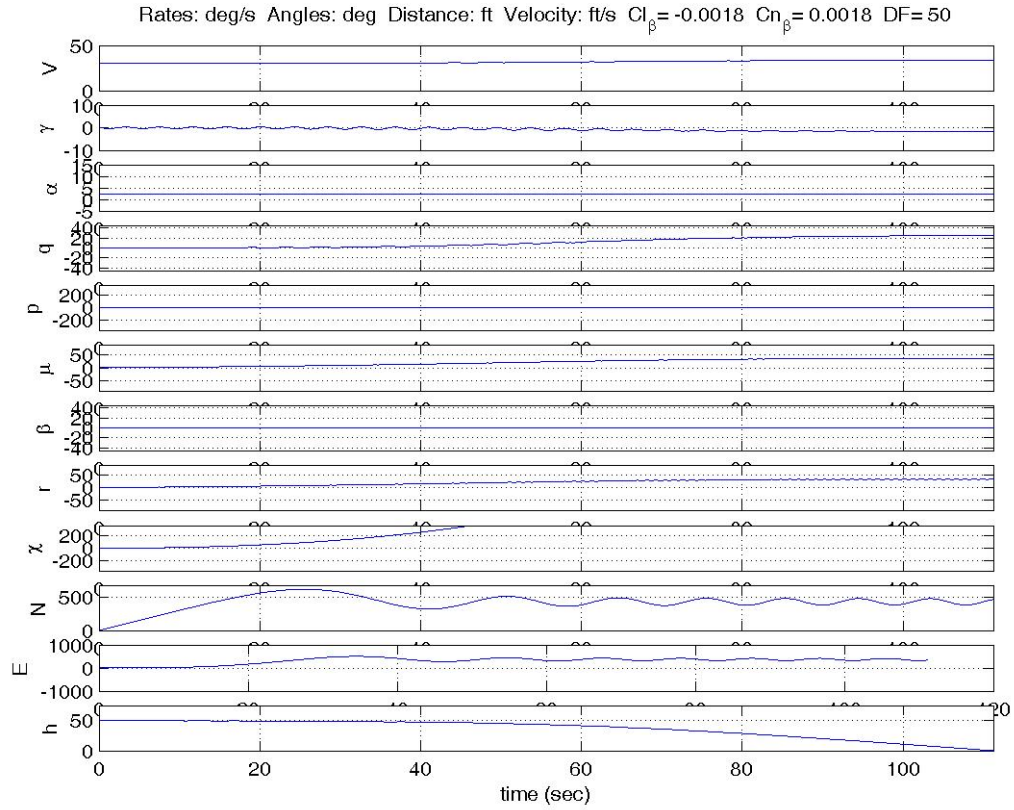


Figure 33. Strip Charts for  $\delta_m=0^\circ$  Bias Removal & Mirroring

These strip charts indicate the yaw rate deviating first followed by a steady right bank and easterly heading change. The aircraft is also behaving steadily enough to recognize what may be the short period oscillation in the flight path angle strip. Tighter axes around previously crucial states are shown in Figure 34.

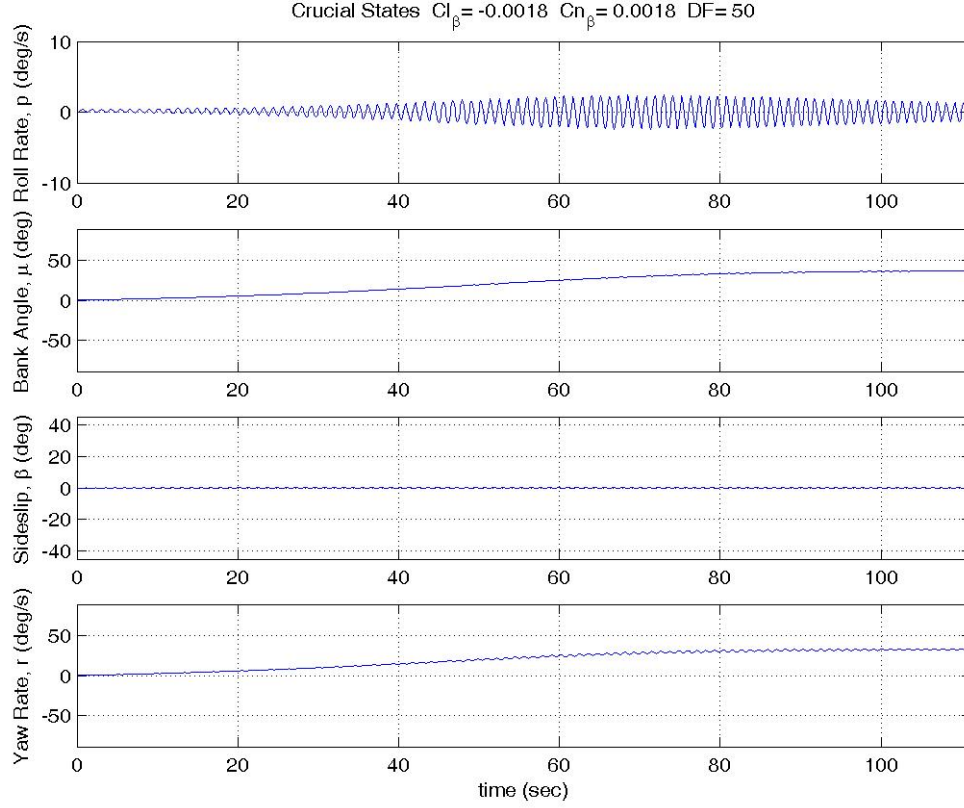


Figure 34. Crucial States for  $\delta_m=0^\circ$  Bias Removal & Mirroring

The roll rate now oscillates symmetrically about zero with approximately  $\pm 2.5$  deg/sec maximum amplitude. It behaves divergently for the first 70 seconds then begins to converge back to zero. Bank angle and yaw rate continue to be coupled with the small amplitude roll rate oscillations leading to eventual 45 deg and 45 deg/s respective values at  $h=0$ . The linearized representation of this modified system is shown in Table 5 along with the modal eigenvalues.

Table 5. A Matrix and Eigenvalues for  $\delta_{rn}=0^\circ$  Bias Removal & Mirroring  
These A,B,C,D matrices represent the linearized version of the BATCAMsim\_g.m  
and BATCAM\_nonlin\_g.mdl for  $\delta_{rn}=0^\circ$  Bias Removal & Mirroring,  $Cn_\beta = +.0018$ , and DF=50

• A Matrix									
ph	V	-0.2351	-32.17	-14.904	0	0	0	-0.000632	0
	$\gamma$	0.07103	0	4.63326	0	0	2.108E-05	0	0
sp	$\alpha$	-0.071	0	-4.6333	1	0	0	0	0
	q	-0.0022	0	-40.9	0	0	0	0	0
roll	p	-3E-18	0	0	0	-8.6302	0	-74.17685	0
	$\mu$	0	-2.11E-05	2.1E-34	0	0.99928	0	-0.004791	0.0379166
dr	$\beta$	-7E-07	0	0	0	0.03792	1.0723333	-0.399961	-0.999281
	r	0.00081	0	0	0	0	0	20.437411	0

Eigenvalues: -0.11756 ± 1.5071i -2.3166 ± 5.961i 0 -0.19998 ± 4.5147i  
Entire A Matrix -8.6302

-9.1672

0.054624 ± 5.2815i

-2.4595 ± 5.8976i

0.027459

0.020093 ± 1.5031i

-9.163

0.052394 ± 5.2824i

0.028046

An evaluation of the eigenvalues for the linearized modified system demonstrates improvement, but remaining instability in the coupled lateral-directional mode. The red text highlights the positive, and therefore, unstable eigenvalues. The reduction of the  $Cl_\beta$  stability coefficient was done after variation of the yaw and damping terms. The large -74.17685 coupling value with the dashed lateral-directional portion of the A matrix indicates a large interaction based on the change in  $\beta$  state. The large number is due to the perturbation of the the system in radians about the trimmed state, and the computation of the force and moment coefficients as per degree units. The final unit solution between the two holds up because the coefficients are non-dimensional. The important factor is that  $Cl_\beta$  is the primary catalyst that begins deviation from SLUF when  $\beta$  is perturbed. Suggestions for continued stability improvement are discussed in Chapter V.

## **V. Conclusions and Recommendations**

### **Chapter Overview**

Overall, the original research objectives for this project were achieved with the exception of the use of dynamic optimization to predict the flight envelope of the aircraft. Empirically based mathematical models now exist for the MAV foldable rotatable tail configuration. An adjustable trimming function is available for SLUF and could readily be adapted for steady level turns. Non-linear EOM modeling has predicted open loop stability characteristics for the bare airframe as well as other artificial, but achievable, modifications. The predicted instabilities match concurrent testing on the V-tail variant of the MAV, and the combined set of Matlab and Simulink code is ready to aid in future steps towards stability.

### **Conclusions of Research**

The current MAV configuration is unstable in roll and spiral modes. This open loop instability repeatedly led back to a review of the original data. The curve fits of experimental data match extremely well using a second order quadratic interaction method. One drawback of the accurate curve fits is the distinct presence of errors and anomalies associated with experimental measurements and rapid prototype models. Therefore some minor adjustments of the input data were necessary for an understanding of potentially achievable stability with the MAV.

The dominant divergent mode is the coupled lateral-directional mode. The primary roll term of interest is the dihedral effect,  $Cl_\beta$ . However, this research has shown

the numerical simulation has a dramatic dependence on damping terms and directional stability,  $Cl_\beta$ , as well.

The variation of  $Cl_\beta$  and  $Cn_\beta$  yielded significant improvements toward stability, but did not create a stable system. The linearized eigenvalues were improved by two orders of magnitude, and the resulting simulated flight time increased by more than 20 times. The varied stability values represent practical modifications to the aircraft that could combine to significantly improve the MAV's stability. The results suggest that enlarged ventral fins or other vertical surfaces combined with less dihedral in the wing would create a better behaved aircraft.

In addition, unaturally high damping factors (50 times the standard estimate) shifted the focus to the coupled lateral directional behavior. Linearization of the complicated system and eigenvalue analysis provide valuable insight to the cause of instability and the relationships between the various modes. Although all but one of the research objectives were met, there is clearly much more required to succeed with controlled maneuvering flight.

### **Recommendations for Future Research**

The following recommendations for future efforts aim either toward improving the stability of the current configuration, or propose alternative design concepts that may more effectively achieve the mission requirements for the ever-changing Global War on Terrorism:

***Reduced Dihedral and Improved Directional Stability.***

The parametric study of  $Cl_\beta$  and  $Cn_\beta$  shows that near-zero dihedral effect (perhaps some anhedral) in conjunction with improved yaw stability from larger ventral fins can approach neutral stability. Further investigation is required to determine if coupled lateral-directional stability can be achieved using the appropriate combination of these terms.

### ***Roll Control.***

The clear roll control deficiency of the current configuration requires further attention. The first action may be to add directional damping terms dependent on tail elevation and rotation that contribute similarly to a rudder or elevator based on their trigonometric contributions. Another variation on the current open loop simulation setup includes expanding the  $\delta_m$  data range to create broader curve fits.

If deeper applications of the current method are unsuccessful, wing roll control methods should be investigated. A prime roll control candidate seems to be some form of wing warping via piezoelectric, cable and servo, or other means. This method would utilize the benefits of the flexible wing design. If such methods are too complicated, spoilers could be used in similar fashion to Hoey's Raven to take advantage of the proverse yaw traits in the absence of significant vertical stabilizer effects.

### ***MIMO Feedback Control.***

The highly coupled modes of this non-linear system will require feedback control for trimmed flight. The highest probability for success most likely is with MIMO feedback control design. Chasing this system by closing one loop at a time would be an exhaustive effort. Successful feedback control for a Stability Augmentation System

(SAS) and then a Control Augmentation System (CAS) would allow smooth incorporation of existing trajectory optimization code for various field deployment scenarios.

### ***Alternate Tail Configuration.***

The rotatable tail configuration of the MAV may prove flyable, but could be too complicated to be viable for field deployment. Another control method to consider is a telescoping V-tail solution for compact portability. A telescoping tail boom mounted on the fuselage with elevons similar to that of previous V-tail configurations may provide compact storage dimensions and the desired handling qualities for the small scale reconnaissance mission. This proposed configuration would need micro servos positioned at the extended end of the tail boom directly attached to the elevons. Coiled wires inside the tail boom could connect the telescoped actuators to the MAV power supply.

Disadvantages associated with a telescoped V-tail configuration may be added weight or insufficient rigidity for proper control authority. However, the potential advantages seem to outweigh most hurdles to the solution. First of all, V-tail configuration are proven capable sufficient control. Control authority deficiencies could easily be improved by lengthening the tail moment arm with very little storage penalties. Finally, the telescoping tail could be air deployable from its storage case. A drogue parachute could extended the tail just prior to extracting the MAV from an airborne transit container. This capability would expand the mission capability to battle damage assessment for strike aircraft.



The MAV's small unit level tactical reconnaissance mission is critical in the Global War On Terrorism. This study has uncovered unique instability due to the lateral-directional coupling of individually stable modes. Key factors in the coupling are  $Cl_\beta$ ,  $Cn_\beta$ , and dynamic rate damping effects. Further study of the combined effects of these lateral-directional terms in specific relation to flexible winged, rotatable tail MAVs is required for successful development of a stable flight demonstrator.

## Appendix A: Experimental Data for Tail Effectiveness

							Lift	Drag	Roll Mom	Pitch Mom	Yaw Mom	Side
Beta	$\delta_e$	$\delta_m$	Tail	Stab	Uoo	alpha_c	$C_L$	$C_D$	$C_{l_r}$	$C_m$	$C_n$	$C_{Yr}$
0	-30	-20	1	1	29.8094	4.7498	1.2057	0.21286	0.00269	-0.0671	-0.0206	0.01789
0	-20	-20	1	1	29.7731	4.76611	1.24259	0.17807	0.00563	-0.1585	-0.0097	-0.0112
0	-10	-20	1	1	29.7066	4.78136	1.27712	0.16489	0.00621	-0.2273	-0.0048	-0.025
0	0	-20	1	1	29.7416	4.81178	1.34594	0.15763	0.00853	-0.3593	0.00299	-0.0471
0	14	-20	1	1	29.8197	4.84565	1.4226	0.17478	0.01054	-0.4975	0.01242	-0.0761
0	-30	-12	1	1	29.8017	4.83001	1.19034	0.21414	0.00445	-0.0342	-0.0108	-0.009
0	-20	-12	1	1	29.7499	4.85204	1.24019	0.17868	0.00606	-0.1496	-0.0068	-0.0194
0	-10	-12	1	1	29.7293	4.8678	1.27584	0.16517	0.00647	-0.2243	-0.0029	-0.0302
0	0	-12	1	1	29.7465	4.89991	1.3485	0.15803	0.00851	-0.3605	0.00261	-0.0462
0	14	-12	1	1	29.864	4.9319	1.42089	0.17347	0.00991	-0.4969	0.0087	-0.0659
0	-30	0	1	1	29.8113	4.8276	1.18488	0.21376	0.00675	-0.0207	-0.0002	-0.0376
0	-20	0	1	1	29.7287	4.85031	1.23626	0.17794	0.0072	-0.1396	-0.0006	-0.0374
0	-10	0	1	1	29.7028	4.86759	1.27537	0.16418	0.00737	-0.2198	0.00148	-0.0431
0	0	0	1	1	29.7564	4.90119	1.3514	0.15734	0.00837	-0.3649	0.00203	-0.0453
0	14	0	1	1	29.845	4.93476	1.42736	0.1724	0.00882	-0.5031	0.00093	-0.0447
0	-30	7	1	1	29.8009	4.82815	1.18613	0.21374	0.00859	-0.0283	0.00875	-0.0616
0	-20	7	1	1	29.7359	4.85044	1.23656	0.17782	0.00835	-0.1411	0.00457	-0.0513
0	-10	7	1	1	29.7114	4.8673	1.27472	0.16443	0.00786	-0.221	0.0043	-0.0512
0	0	7	1	1	29.7517	4.89991	1.34851	0.15684	0.00826	-0.3596	0.0014	-0.0438
0	14	7	1	1	29.8023	4.93432	1.42638	0.17154	0.00799	-0.4987	-0.0035	-0.0322
0	-30	14	1	1	29.7853	4.83239	1.19572	0.21345	0.01023	-0.0545	0.01737	-0.0846
0	-20	14	1	1	29.7561	4.85026	1.23616	0.17787	0.00927	-0.1483	0.01005	-0.0662
0	-10	14	1	1	29.7188	4.86692	1.27385	0.16511	0.00841	-0.2215	0.00751	-0.0601
0	0	14	1	1	29.7536	4.89868	1.34573	0.15649	0.00821	-0.3546	0.00115	-0.0435
0	14	14	1	1	29.7692	4.93298	1.42333	0.17013	0.00749	-0.4902	-0.0068	-0.0233
0	-30	25	1	1	29.7774	4.83563	1.20305	0.2143	0.01219	-0.0814	0.0253	-0.1057
0	-20	25	1	1	29.8224	4.84988	1.2353	0.17863	0.01008	-0.1562	0.0145	-0.0785
0	-10	25	1	1	29.7123	4.86684	1.27366	0.16623	0.00887	-0.2192	0.00971	-0.0663
0	0	25	1	1	29.737	4.89758	1.34323	0.15642	0.0083	-0.3489	0.001	-0.0434
0	14	25	1	1	29.7877	4.92834	1.41284	0.16789	0.00705	-0.4742	-0.0096	-0.0157

## Appendix B: Tail Effectiveness Curve Fit Matlab Code

```
0001 % Capt Travis Higgs
0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Drag Coefficient, CD, results.
0006 % CD_polyfit_2d.m
0007
0008 % Version d eliminates bias to place the appropriate value at the (0,0)
0009 % point on the surface plot
0010
0011 clear; clc; %clf;
0012 format short g
0013
0014 % pull data from appropriate matrix
0015 load data2b.prn
0016
0017 % variable definitions:
0018 delta_e = data2b(:,2); % tail deflection (deg)
0019 alpha = data2b(:,10); % angle of attack (deg)
0020 delta_rn = data2b(:,3); % tail rotation (deg)
0021 beta = data2b(:,1); % side slip angle (deg)
0022
0023 delta_er = data2b(:,2)*pi/180; % tail deflection (rad)
0024 alphas = data2b(:,10)*pi/180; % angle of attack (rad)
0025 delta_rnr = data2b(:,3)*pi/180; % tail rotation (rad)
0026 betas = data2b(:,1)*pi/180; % side slip angle (rad)
0027
0028 CD = data2b(:,12); % Coefficient of Lift
0029
0030 % create pieces that will form large A matrix that will build a fit function
0031 % of the form Ax=b:
0032 % Polynomial Fit instead of using sin and cos...
0033 % x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0034 % +x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0035 m = length(data2b);
0036 % Calculate the lead coefficient (A1p) based on alpha without tail deflections
0037 A1p = ones(m,1)
0038 A2p = delta_e;
0039 A3p = delta_e.^2;
0040 A4p = delta_rn;
0041 A5p = delta_rn.^2;
0042 A6p = delta_e.*delta_rn;
0043 A7p = delta_e.^2.*delta_rn;
0044 A8p = delta_e.*delta_rn.^2;
0045 A9p = delta_e.^2.*delta_rn.^2;
0046
0047 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p]
0048
0049 CDalpha = 3E-05*alpha.^3+0.0008*alpha.^2+0.0052*alpha+0.078*ones(m,1); % (R^2=0.9997)
0050 bp = CD-CDalpha;
0051
```

```

0052 xp=Ap\bp % \ is the least squares operator for Matlab
0053
0054 CD_polyfit = Ap*xp+CDalpha;
0055
0056 % diff = CD-CD_polyfit_2c;
0057 % perc_diff = diff./CD*100;
0058 % Compare = [delta_rn*180/pi delta_e*180/pi CD CD_polyfit_2c diff perc_diff]
0059 %
0060 save CD_polyfit_2d xp;
0061
0062 %plot surface contours for comparison
0063 %Data:
0064 Xd=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0065 Yd=delta_e(1:5); %5 delta_e settings
0066 Zd=[CD(1:5) CD(6:10) CD(11:15) CD(16:20) CD(21:25) CD(26:30)]; %stacks the CD data into a
5x6 matrix
0067 %Establish the axes bounds
0068 lbdn = min(delta_rn); %lower bound delta_rn
0069 ubdn = max(delta_rn); %upper bound delta_rn
0070 lbde = min(delta_e); %lower bound delta_e
0071 ubde = max(delta_e); %upper bound delta_e
0072 zmin=min(CD); %lower bound z-axis
0073 zmax=max(CD); %upper bound z-axis
0074 figure(20)
0075 subplot(2,2,1)
0076 hold on
0077 contourf(Xd,Yd,Zd)
0078 plot(0,0,'+k')
0079 hold off
0080 axis([lbdn ubdn lbde ubde])
0081
0082 caxis([zmin zmax])
0083 colormap jet
0084 %colorbar
0085 xlabel('\delta_r_n (deg)')
0086 ylabel('\delta_e (deg)')
0087 title('C_D Experimental Data')
0088
0089 subplot(2,2,3)
0090 surf(Xd,Yd,Zd)
0091 caxis([zmin zmax])
0092 colormap jet
0093 %colorbar
0094 axis([-15 25 -30 30 zmin zmax])
0095 xlabel('\delta_r_n (deg)')
0096 ylabel('\delta_e (deg)')
0097 zlabel('C_D')
0098
0099 %Polynomial Fit:
0100 Xfp=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0101 Yfp=delta_e(1:5); % 5 delta_e settings

```

```

0102 Zfp=[CD_polyfit(1:5) CD_polyfit(6:10) CD_polyfit(11:15) CD_polyfit(16:20) CD_polyfit(21:25)
CD_polyfit(26:30)];
0103 subplot(2,2,2)
0104 hold on
0105 contourf(Xfp,Yfp,Zfp)
0106 plot(0,0,'+k')
0107 hold off
0108 axis([lbdrn ubdrn lbde ubde])
0109 caxis([zmin zmax])
0110 colormap jet
0111 colorbar
0112 xlabel('\delta_r_n (deg)')
0113 ylabel('\delta_e (deg)')
0114 title('C_D Polynomial Fit','Color','r')
0115
0116 subplot(2,2,4)
0117 surf(Xfp,Yfp,Zfp)
0118 caxis([zmin zmax])
0119 colormap jet
0120 colorbar
0121 axis([-15 25 -30 30 zmin zmax])
0122 xlabel('\delta_r_n (deg)')
0123 ylabel('\delta_e (deg)')
0124 zlabel('C_D')
0125 f=3;
0126 title(['C_D_w(\alpha)+' ,...
0127     num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0128     '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0129     '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0130     '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...
0131     num2str(xp(9),f),'*\delta_e^2*\delta_r_n^2 (deg)
'],'Color','r','FontSize',6)
0132
0133
0134 saveas(20,'CD_polyfit_2d.jpg')
0135
0136 %Output from this code is as follows:

```

```

0001 % Capt Travis Higgs

0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Drag Coefficient, CD, results.
0006 % CD_polyfit_2d.m
0007
0008 % Version d eliminates bias to place the appropriate value at the (0,0)
0009 % point on the surface plot
0010
0011 clear; clc; %clf;
0012 format short g
0013
0014 %pull data from appropriate matrix
0015 load data2b.prn
0016
0017 %variable definitions:
0018 delta_e = data2b(:,2); % tail deflection (deg)
0019 alpha = data2b(:,10); % angle of attack (deg)
0020 delta_rn = data2b(:,3); % tail rotation (deg)
0021 beta = data2b(:,1); % side slip angle (deg)
0022
0023 delta_er = data2b(:,2)*pi/180; % tail deflection (rad)
0024 alphas = data2b(:,10)*pi/180; % angle of attack (rad)
0025 delta_rnr = data2b(:,3)*pi/180; % tail rotation (rad)
0026 betas = data2b(:,1)*pi/180; % side slip angle (rad)
0027
0028 CD = data2b(:,12); % Coefficient of Lift
0029
0030 %create pieces that will form large A matrix that will build a fit function
0031 %of the form Ax=b:
0032 %Polynomial Fit instead of using sin and cos...
0033 %x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0034 %+x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0035 m = length(data2b);
0036 %Calculate the lead coefficient (A1p) based on alpha without tail deflections
0037 A1p = ones(m,1)
0038 A2p = delta_e;
0039 A3p = delta_e.^2;
0040 A4p = delta_rn;
0041 A5p = delta_rn.^2;
0042 A6p = delta_e.*delta_rn;
0043 A7p = delta_e.^2.*delta_rn;
0044 A8p = delta_e.*delta_rn.^2;
0045 A9p = delta_e.^2.*delta_rn.^2;
0046
0047 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p]
0048
0049 CDalpha = 3E-05*alpha.^3+0.0008*alpha.^2+0.0052*alpha+0.078*ones(m,1); %(R^2=0.9997)
0050 bp = CD-CDalpha;
0051
0052 xp=Ap\bp % \ is the least squares operator for Matlab

```

```

0053
0054 CD_polyfit = Ap*xp+CDalpha;
0055
0056 % diff = CD-CD_polyfit_2c;
0057 % perc_diff = diff./CD*100;
0058 % Compare = [delta_rn*180/pi delta_e*180/pi CD CD_polyfit_2c diff perc_diff]
0059 %
0060 save CD_polyfit_2d xp;
0061
0062 %plot surface contours for comparison
0063 %Data:
0064 Xd=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0065 Yd=delta_e(1:5); %5 delta_e settings
0066 Zd=[CD(1:5) CD(6:10) CD(11:15) CD(16:20) CD(21:25) CD(26:30)]; %stacks the CD data into a
5x6 matrix
0067 %Establish the axes bounds
0068 lbdrn = min(delta_rn); %lower bound delta_rn
0069 ubdrn = max(delta_rn); %upper bound delta_rn
0070 lbde = min(delta_e); %lower bound delta_e
0071 ubde = max(delta_e); %upper bound delta_e
0072 zmin=min(CD); %lower bound z-axis
0073 zmax=max(CD); %upper bound z-axis
0074 figure(20)
0075 subplot(2,2,1)
0076 hold on
0077 contourf(Xd,Yd,Zd)
0078 plot(0,0,'+k')
0079 hold off
0080 axis([lbdrn ubdrn lbde ubde])
0081
0082 caxis([zmin zmax])
0083 colormap jet
0084 %colorbar
0085 xlabel('\delta_r_n (deg)')
0086 ylabel('\delta_e (deg)')
0087 title('C_D Experimental Data')
0088
0089 subplot(2,2,3)
0090 surf(Xd,Yd,Zd)
0091 caxis([zmin zmax])
0092 colormap jet
0093 %colorbar
0094 axis([-15 25 -30 30 zmin zmax])
0095 xlabel('\delta_r_n (deg)')
0096 ylabel('\delta_e (deg)')
0097 zlabel('C_D')
0098
0099 %Polynomial Fit:
0100 Xfp=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0101 Yfp=delta_e(1:5); % 5 delta_e settings

```

```

0102 Zfp=[CD_polyfit(1:5) CD_polyfit(6:10) CD_polyfit(11:15) CD_polyfit(16:20) CD_polyfit(21:25)
CD_polyfit(26:30)];
0103 subplot(2,2,2)
0104 hold on
0105 contourf(Xfp,Yfp,Zfp)
0106 plot(0,0,'+k')
0107 hold off
0108 axis([lbdrn ubdrn lbde ubde])
0109 caxis([zmin zmax])
0110 colormap jet
0111 colorbar
0112 xlabel('\delta_r_n (deg)')
0113 ylabel('\delta_e (deg)')
0114 title('C_D Polynomial Fit','Color','r')
0115
0116 subplot(2,2,4)
0117 surf(Xfp,Yfp,Zfp)
0118 caxis([zmin zmax])
0119 colormap jet
0120 colorbar
0121 axis([-15 25 -30 30 zmin zmax])
0122 xlabel('\delta_r_n (deg)')
0123 ylabel('\delta_e (deg)')
0124 zlabel('C_D')
0125 f=3;
0126 title(['C_D_w(\alpha)+' ,...
0127     num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0128     '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0129     '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0130     '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...
0131     num2str(xp(9),f),'*\delta_e^2*\delta_r_n^2 (deg)
'],'Color','r','FontSize',6)
0132
0133
0134 saveas(20,'CD_polyfit_2d.jpg')
0135
0136 %Output from this code is as follows:

```



```

0001 % Capt Travis Higgs
0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Drag Coefficient, CD, results.
0006 % CD_polyfit_2d.m
0007
0008 % Version d eliminates bias to place the appropriate value at the (0,0)
0009 % point on the surface plot
0010
0011 clear; clc; %clf;
0012 format short g
0013
0014 %pull data from appropriate matrix
0015 load data2b.prn
0016
0017 %variable definitions:
0018 delta_e = data2b(:,2); % tail deflection (deg)
0019 alpha = data2b(:,10); % angle of attack (deg)
0020 delta_rn = data2b(:,3); % tail rotation (deg)
0021 beta = data2b(:,1); % side slip angle (deg)
0022
0023 delta_er = data2b(:,2)*pi/180; % tail deflection (rad)
0024 alphas = data2b(:,10)*pi/180; % angle of attack (rad)
0025 delta_rnr = data2b(:,3)*pi/180; % tail rotation (rad)
0026 betar = data2b(:,1)*pi/180; % side slip angle (rad)
0027
0028 CD = data2b(:,12); % Coefficient of Lift
0029
0030 %create pieces that will form large A matrix that will build a fit function
0031 %of the form Ax=b:
0032 %Polynomial Fit instead of using sin and cos...
0033 %x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0034 %+x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0035 m = length(data2b);
0036 %Calculate the lead coefficient (A1p) based on alpha without tail deflections
0037 A1p = ones(m,1)
0038 A2p = delta_e;
0039 A3p = delta_e.^2;
0040 A4p = delta_rn;
0041 A5p = delta_rn.^2;
0042 A6p = delta_e.*delta_rn;
0043 A7p = delta_e.^2.*delta_rn;
0044 A8p = delta_e.*delta_rn.^2;
0045 A9p = delta_e.^2.*delta_rn.^2;
0046
0047 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p]
0048
0049 CDalpha = 3E-05*alpha.^3+0.0008*alpha.^2+0.0052*alpha+0.078*ones(m,1); % (R^2=0.9997)
0050 bp = CD-CDalpha;
0051
0052 xp=Ap\bp % \ is the least squares operator for Matlab
0053

```

```

0054 CD_polyfit = Ap*xp+CDalpha;
0055
0056 % diff = CD-CD_polyfit_2c;
0057 % perc_diff = diff./CD*100;
0058 % Compare = [delta_rn*180/pi delta_e*180/pi CD CD_polyfit_2c diff perc_diff]
0059 %
0060 save CD_polyfit_2d xp;
0061
0062 %plot surface contours for comparison
0063 %Data:
0064 Xd=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0065 Yd=delta_e(1:5); %5 delta_e settings
0066 Zd=[CD(1:5) CD(6:10) CD(11:15) CD(16:20) CD(21:25) CD(26:30)]; %stacks the CD data into a
5x6 matrix
0067 %Establish the axes bounds
0068 lbdrn = min(delta_rn); %lower bound delta_rn
0069 ubdrn = max(delta_rn); %upper bound delta_rn
0070 lbde = min(delta_e); %lower bound delta_e
0071 ubde = max(delta_e); %upper bound delta_e
0072 zmin=min(CD); %lower bound z-axis
0073 zmax=max(CD); %upper bound z-axis
0074 figure(20)
0075 subplot(2,2,1)
0076 hold on
0077 contourf(Xd,Yd,Zd)
0078 plot(0,0,'+k')
0079 hold off
0080 axis([lbdrn ubdrn lbde ubde])
0081
0082 caxis([zmin zmax])
0083 colormap jet
0084 %colorbar
0085 xlabel('\delta_r_n (deg)')
0086 ylabel('\delta_e (deg)')
0087 title('C_D Experimental Data')
0088
0089 subplot(2,2,3)
0090 surf(Xd,Yd,Zd)
0091 caxis([zmin zmax])
0092 colormap jet
0093 %colorbar
0094 axis([-15 25 -30 30 zmin zmax])
0095 xlabel('\delta_r_n (deg)')
0096 ylabel('\delta_e (deg)')
0097 zlabel('C_D')
0098
0099 %Polyniomial Fit:
0100 Xfp=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0101 Yfp=delta_e(1:5); % 5 delta_e settings
0102 Zfp=[CD_polyfit(1:5) CD_polyfit(6:10) CD_polyfit(11:15) CD_polyfit(16:20) CD_polyfit(21:25)
CD_polyfit(26:30)];

```

```

0103 subplot(2,2,2)
0104 hold on
0105 contourf(Xfp,Yfp,Zfp)
0106 plot(0,0,'+k')
0107 hold off
0108 axis([lbdrn ubdrn lbde ubde])
0109 caxis([zmin zmax])
0110 colormap jet
0111 colorbar
0112 xlabel('\delta_r_n (deg)')
0113 ylabel('\delta_e (deg)')
0114 title('C_D Polynomial Fit','Color','r')
0115
0116 subplot(2,2,4)
0117 surfc(Xfp,Yfp,Zfp)
0118 caxis([zmin zmax])
0119 colormap jet
0120 colorbar
0121 axis([-15 25 -30 30 zmin zmax])
0122 xlabel('\delta_r_n (deg)')
0123 ylabel('\delta_e (deg)')
0124 zlabel('C_D')
0125 f=3;
0126 title(['C_D_w(\alpha)+',...
0127     num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0128     '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0129     '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0130     '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...
0131     num2str(xp(9),f),'*\delta_e^2*\delta_r_n^2 (deg)
'],'Color','r','FontSize',6)
0132
0133
0134 saveas(20,'CD_polyfit_2d.jpg')
0135
0136 %Output from this code is as follows:

```

```

0001 % Capt Travis Higgs

0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Rolling Moment Coefficient, C_l, results.
0006 % C_l_polyfit_2e.m
0007
0008 clear; clc; clf;
0009 format short g
0010
0011 %Version e uses massaged data for a zeroed and symmetric Cl map about the
0012 %delta_rn axis
0013
0014 %pull data from appropriate matrix
0015 %NOTE:the data must be sorted so that the x axis data is in ASCENDING order
0016 %if the countourf or surfc functions are used (MATlab R13)
0017 load data2d.prn %This data has been zeroed at delta_rn=0 and the (+) delta_rn measurements
0018 %have been mirrored to remove assymetry about the delta_rn axis
0019 % load Clwab4.prn
0020
0021 %variable definitions:
0022 % alphaw = Clwab4(:,2); % Wing angle of attack (deg)
0023 % alphawp = Clwab4(:,3); % Wing angle of attack (deg)
0024 % betaw = Clwab4(:,4); % Wing side slip angle (deg)
0025 % C_lw = Clwab4(:,6); % Wing Rolling Moment Coefficient
0026
0027 delta_e = data2d(:,2); % tail deflection (deg)
0028 alpha = data2d(:,10); % angle of attack (deg)
0029 delta_rn = data2d(:,3); % tail rotation (deg)
0030 beta = data2d(:,1); % side slip angle (deg)
0031
0032 delta_er = data2d(:,2)*pi/180; % tail deflection (rad)
0033 alphas = data2d(:,10)*pi/180; % angle of attack (rad)
0034 delta_rnr = data2d(:,3)*pi/180; % tail rotation (rad)
0035 betar = data2d(:,1)*pi/180; % side slip angle (rad)
0036
0037 m = length(data2d);
0038 %bias = data2d(14,14)
0039 C_l = data2d(:,14);%-bias*ones(m,1); % A/C Rolling Moment Coefficient with bias at 0,0
removed
0040
0041 %Polynomial Fit instead of using sin and cos...
0042 %This is C_l for the WHOLE AIRCRAFT where x1=the function found for C_lw
0043 % x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0044 % +x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0045
0046 A1p = ones(m,1);
0047 A2p = delta_e;
0048 A3p = delta_e.^2;
0049 A4p = delta_rn;
0050 A5p = delta_rn.^2;
0051 A6p = delta_e.*delta_rn;

```

```

0052 A7p = delta_e.^2.*delta_rn;
0053 A8p = delta_e.*delta_rn.^2;
0054 A9p = delta_e.^2.*delta_rn.^2;
0055
0056 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p];
0057
0058 %Simplified C_l_beta function (disregard alpha)
0059 Clb=-.0018; %Dihedral Effect of Wing
0060 C_lbeta = Clb*beta;
0061 bp = C_l-C_lbeta;
0062
0063 xp=Ap\bp % \ is the least squares operator for Matlab
0064
0065 C_l_polyfit = Ap*xp+C_lbeta;
0066 save C_l_polyfit_2e xp Clb;
0067
0068 %%%%%%%%%%%%%%%plot surface contours for comparison%%%%%%%%%%%%%%
0069 %Data:
0070 ndrn=7; %number of discrete delta_rn settings in data set
0071 nde=5; %number of discrete delta_e settings in data set
0072
0073 Xd=[delta_rn(1)];
0074 Xd=[delta_rn(1)];
0075 for i=1:(ndrn-1)
0076     Xd=[Xd delta_rn(1+i*nde)];
0077 end
0078 Yd=[delta_e(1:nde)]; %
0079 Zd=[C_l(1:nde)];
0080 for i=1:(ndrn-1)
0081     Zd=[Zd C_l((1+i*nde):((i+1)*nde))];
0082 end
0083 %Establish the axes bounds
0084 lbdrn = min(delta_rn); %lower bound delta_rn
0085 ubdrn = max(delta_rn); %upper bound delta_rn
0086 lbde = min(delta_e); %lower bound delta_e
0087 ubde = max(delta_e); %upper bound delta_e
0088 zmin = min(C_l); %lower bound z-axis
0089 zmax = max(C_l); %upper bound z-axis
0090 n = 20; %number of contour lines in the surface plot
0091 figure(11)
0092 clf;
0093 subplot(2,2,1)
0094 hold on
0095 contourf(Xd,Yd,Zd,n)
0096 plot(0,0,'+k')
0097 hold off
0098 axis([lbdrn ubdrn lbde ubde])
0099 caxis([zmin zmax])
0100 colormap jet
0101 %colorbar
0102 xlabel('\delta_r_n (deg)')
0103 ylabel('\delta_e (deg)')
0104 title('C_l Experimental Data')

```

```

0105
0106 subplot(2,2,3)
0107 surf(Xd,Yd,Zd)
0108 caxis([zmin zmax])
0109 colormap jet
0110 %colorbar
0111 axis([lbdrn ubdrn lbde ubde zmin zmax])
0112 xlabel('\delta_r_n (deg)')
0113 ylabel('\delta_e (deg)')
0114 zlabel('C_l')
0115
0116 %Polyniomial Fit:
0117 Xfp=[delta_rn(1)];
0118 Xfp=[delta_rn(1)];
0119 for i=1:(ndrn-1)
0120     Xfp=[Xfp delta_rn(1+i*nde)];
0121 end
0122 Yfp=[delta_e(1:nde)]; %
0123 Zfp=[C_l(1:nde)];
0124 for i=1:(ndrn-1)
0125     Zfp=[Zfp C_l((1+i*nde):((i+1)*nde))];
0126 end
0127
0128 subplot(2,2,2)
0129 hold on
0130 contourf(Xfp,Yfp,Zfp,n)
0131 plot(0,0,'+k')
0132 hold off
0133 axis([lbdrn ubdrn lbde ubde])
0134 caxis([zmin zmax])
0135 colormap jet
0136 colorbar
0137 xlabel('\delta_r_n (deg)')
0138 ylabel('\delta_e (deg)')
0139 title('C_l Polynomial Fit','Color','r')
0140 text(lbdrn-25,ubde+5,['Mirrored About \delta_r_n'],'Color','b','FontSize',10)
0141
0142 subplot(2,2,4)
0143 surf(Xfp,Yfp,Zfp)
0144 caxis([zmin zmax])
0145 colormap jet
0146 colorbar
0147 axis([lbdrn ubdrn lbde ubde zmin zmax])
0148 xlabel('\delta_r_n (deg)')
0149 ylabel('\delta_e (deg)')
0150 zlabel('C_l')
0151 %Show equation for Aircraft C_l
0152 f = 3;
0153 title(['C_l_w(\beta)+',...
0154     num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0155     '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0156     '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0157     '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...

```

```

0158     num2str(xp(9),f),...
0159     '*\delta_e^2*\delta_r_n^2 (deg)
    '], 'Color', 'r', 'FontSize', 6)
0160
0161 saveas(11, 'C_1_polyfit_2e.jpg')
0162
0163 %Output from this code is as follows:

```

```

0001 % Capt Travis Higgs

0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Yawing Moment Coefficient, Cn, results.
0006 % Cn_polyfit_2d.m
0007
0008 % Version d eliminates bias to place the appropriate value at the (0,0)
0009 % point on the surface plot
0010
0011 clear; clc; %clf;
0012 format short g
0013
0014 %pull data from appropriate matrix
0015 %load data2e.prm %raw data resorted for Matlab compatability
0016 load data2e.prm %all Cn values at delta_rn=0 are equal (0 with bias removed)
0017
0018 %variable definitions:
0019 delta_e = data2e(:,2); % tail deflection (deg)
0020 alpha = data2e(:,10); % angle of attack (deg)
0021 delta_rn = data2e(:,3); % tail rotation (deg)
0022 beta = data2e(:,1); % side slip angle (deg)
0023
0024 delta_er = data2e(:,2)*pi/180; % tail deflection (rad)
0025 alphas = data2e(:,10)*pi/180; % angle of attack (rad)
0026 delta_rnr = data2e(:,3)*pi/180; % tail rotation (rad)
0027 betar = data2e(:,1)*pi/180; % side slip angle (rad)
0028
0029
0030 m = length(data2e);
0031 bias = data2e(14,15)
0032 Cn = data2e(:,15)-bias*ones(m,1); % Coefficient of Lift with bias at 0,0 removed
0033
0034 %Polynomial Fit instead of using sin and cos...
0035 %x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0036 %+x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0037 m = length(data2e);
0038 A1p = ones(m,1);
0039 A2p = delta_e;
0040 A3p = delta_e.^2;
0041 A4p = delta_rn;
0042 A5p = delta_rn.^2;
0043 A6p = delta_e.*delta_rn;
0044 A7p = delta_e.^2.*delta_rn;
0045 A8p = delta_e.*delta_rn.^2;
0046 A9p = delta_e.^2.*delta_rn.^2;
0047
0048 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p]
0049
0050 %Cnbeta = -0.0005*beta+0.0019*ones(m,1); % (R^2=.9921)
0051 %Bias of +.0019 removed from Cnbeta curve:
0052 %Cnbeta = -0.0005*beta;

```



```

0053 % Cn(beta) of the aircraft with zero tail deflections
0054 % is artificially increased in slope from -.0005 to .002
0055 Cnb = 0.0018;
0056 Cnbeta = Cnb*beta;
0057 bp = Cn-Cnbeta;
0058
0059 xp=Ap\bp % \ is the least squares operator for Matlab
0060
0061 Cn_polyfit = Ap*xp+Cnbeta;
0062
0063 % diff = Cn-Cn_polyfit;
0064 % perc_diff = diff./Cn*100;
0065 % Compare = [delta_rn*180/pi delta_e*180/pi Cn Cn_polyfit diff perc_diff]
0066 save Cn_polyfit_2d xp Cnb;
0067
0068 %plot surface contours for comparison
0069 %Data:
0070 Xd=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0071 Yd=delta_e(1:5); % 5 delta_e settings
0072 Zd=[Cn(1:5) Cn(6:10) Cn(11:15) Cn(16:20) Cn(21:25) Cn(26:30)]; %stacks the Cn data into a 5x6
matrix
0073 %Establish the axes bounds
0074 lbdrn = min(delta_rn); %lower bound delta_rn
0075 ubdrn = max(delta_rn); %upper bound delta_rn
0076 lbde = min(delta_e); %lower bound delta_e
0077 ubde = max(delta_e); %upper bound delta_e
0078 n = 20; %number of contour lines in the surface plot
0079 zmin=min(Cn);
0080 zmax=max(Cn);
0081 figure(50)
0082 subplot(2,2,1)
0083 hold on
0084 contourf(Xd,Yd,Zd,n)
0085 plot(0,0,'+k')
0086 hold off
0087 axis([lbdrn ubdrn lbde ubde])
0088 caxis([zmin zmax])
0089 colormap jet
0090 %colorbar
0091 xlabel('\delta_r_n (deg)')
0092 ylabel('\delta_e (deg)')
0093 title('C_n Experimental Data')
0094
0095 subplot(2,2,3)
0096 surfc(Xd,Yd,Zd)
0097 caxis([zmin zmax])
0098 colormap jet
0099 %colorbar
0100 axis([-15 25 -30 30 zmin zmax])
0101 xlabel('\delta_r_n (deg)')
0102 ylabel('\delta_e (deg)')
0103 zlabel('C_n')

```

```

0104
0105 %Polyniomial Fit:
0106 Xfp=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0107 Yfp=delta_e(1:5); % 5 detla_e settings
0108 Zfp=[Cn_polyfit(1:5) Cn_polyfit(6:10) Cn_polyfit(11:15) Cn_polyfit(16:20) Cn_polyfit(21:25)
Cn_polyfit(26:30)];
0109 subplot(2,2,2)
0110 hold on
0111 contourf(Xfp,Yfp,Zfp,n)
0112 plot(0,0,'+k')
0113 hold off
0114 axis([lbdrn ubdrn lbde ubde])
0115 caxis([zmin zmax])
0116 colormap jet
0117 colorbar
0118 xlabel('\delta_r_n (deg)')
0119 ylabel('\delta_e (deg)')
0120 title('C_n Polynomial Fit','Color','r')
0121 text(lbdrn-25,ubde+5,['Bias = ',num2str(bias),' removed'],'Color','b','FontSize',10)
0122
0123 subplot(2,2,4)
0124 surf(Xfp,Yfp,Zfp)
0125 caxis([zmin zmax])
0126 colormap jet
0127 colorbar
0128 axis([-15 25 -30 30 zmin zmax])
0129 xlabel('\delta_r_n (deg)')
0130 ylabel('\delta_e (deg)')
0131 zlabel('C_n')
0132 f = 3; %number of significant digits
0133 title(['C_n_w(\beta)+',...
0134 num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0135 '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0136 '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0137 '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...
0138 num2str(xp(9),f),'*\delta_e^2*\delta_r_n^2 (deg)
'],'Color','r','FontSize',6)
0139
0140 saveas(50,'Cn_polyfit_2d.jpg')
0141
0142 %Output from this code is as follows:
0001 % Capt Travis Higgs
0002 % Thesis: Batcam Control Law Development
0003 % This function uses least squares optimization
0004 % of MatLab's optimization toolbox to curve fit delta_e and delta_rn
0005 % inputs to corresponding Aircraft Side Force Coefficient, Cy, results.
0006 % Cy_polyfit_2d.m
0007
0008 %Version d eliminates bias to place the appropriate value at the (0,0)
0009 %point on the surface plot
0010
0011 clear; clc; %clf;

```

```

0012 format short g
0013
0014 %pull data from appropriate matrix
0015 %load data2b.prn %raw data resorted for Matlab compatability
0016 load data2e.prn %all Cy values at delta_rn=0 are equal (0 with bias removed)
0017
0018 %variable definitions:
0019 delta_e = data2e(:,2); % tail deflection (deg)
0020 alpha = data2e(:,10); % angle of attack (deg)
0021 delta_rn = data2e(:,3); % tail rotation (deg)
0022 beta = data2e(:,1); % side slip angle (deg)
0023
0024 delta_er = data2e(:,2)*pi/180; % tail deflection (rad)
0025 alphas = data2e(:,10)*pi/180; % angle of attack (rad)
0026 delta_rnr = data2e(:,3)*pi/180; % tail rotation (rad)
0027 betar = data2e(:,1)*pi/180; % side slip angle (rad)
0028
0029 m = length(data2e);
0030 bias = data2e(14,16)
0031 Cy = data2e(:,16)-bias*ones(m,1); % Sideforce Coefficient with bias at 0,0 removed
0032
0033 %Polynomial Fit instead of using sin and cos...
0034 %x1+x2*delta_e+x3*delta_e^2+x4*delta_rn+x5*delta_rn^2+x6*delta_e*delta_rn...
0035 %+x7*delta_e^2*delta_rn+x8*delta_e*delta_rn^2+x9*delta_e^2*delta_rn^2
0036 m = length(data2e);
0037 A1p = ones(m,1);
0038 A2p = delta_e;
0039 A3p = delta_e.^2;
0040 A4p = delta_rn;
0041 A5p = delta_rn.^2;
0042 A6p = delta_e.*delta_rn;
0043 A7p = delta_e.^2.*delta_rn;
0044 A8p = delta_e.*delta_rn.^2;
0045 A9p = delta_e.^2.*delta_rn.^2;
0046
0047 Ap = [A1p A2p A3p A4p A5p A6p A7p A8p A9p]
0048
0049 %Cybeta = 0.0074*beta-0.0455*ones(m,1); %(R^2=.9984)
0050 %Remove Cy bias of -.0455 from the beta fit to make it cross at 0,0.
0051 Cybeta = -0.0074*beta; %(R^2=.9984)
0052
0053 bp = Cy-Cybeta;
0054
0055 xp=Ap\bp % \ is the least squares operator for Matlab
0056
0057 x1 = xp(1)*ones(m,1); %creates the "offset value"
0058 Cy_polyfit = Ap*xp+Cybeta; %subtracts offset value for zero moment at zero inputs
0059
0060 save Cy_polyfit_2d xp;
0061
0062 %plot surface contours for comparison
0063 %Data:

```

```

0064 Xd=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0065 Yd=delta_e(1:5); % 5 detla_e settings
0066 Zd=[Cy(1:5) Cy(6:10) Cy(11:15) Cy(16:20) Cy(21:25) Cy(26:30)]; %stacks the Cy data into a 5x6
matrix
0067 %Establish the axes bounds
0068 lbdrn = min(delta_rn); %lower bound delta_rn
0069 ubdrn = max(delta_rn); %upper bound delta_rn
0070 lbde = min(delta_e); %lower bound delta_e
0071 ubde = max(delta_e); %upper bound delta_e
0072 n = 20; %number of contour lines in the surface plot
0073 zmin=min(Cy);
0074 zmax=max(Cy);
0075 figure(60)
0076 subplot(2,2,1)
0077 hold on
0078 contourf(Xd,Yd,Zd,n)
0079 plot(0,0,'+k')
0080 hold off
0081 axis([lbdrn ubdrn lbde ubde])
0082 caxis([zmin zmax])
0083 colormap jet
0084 %colorbar
0085 xlabel('\delta_r_n (deg)')
0086 ylabel('\delta_e (deg)')
0087 title('C_y Experimental Data')
0088
0089 subplot(2,2,3)
0090 surf(Xd,Yd,Zd)
0091 caxis([zmin zmax])
0092 colormap jet
0093 %colorbar
0094 axis([-15 25 -30 30 zmin zmax])
0095 xlabel('\delta_r_n (deg)')
0096 ylabel('\delta_e (deg)')
0097 zlabel('C_y')
0098
0099 %Polyniomial Fit:
0100 Xfp=[delta_rn(1) delta_rn(6) delta_rn(11) delta_rn(16) delta_rn(21) delta_rn(26)]; %6 delta_rn
settings
0101 Yfp=delta_e(1:5); % 5 detla_e settings
0102 Zfp=[Cy_polyfit(1:5) Cy_polyfit(6:10) Cy_polyfit(11:15) Cy_polyfit(16:20) Cy_polyfit(21:25)
Cy_polyfit(26:30)];
0103 subplot(2,2,2)
0104 hold on
0105 contourf(Xfp,Yfp,Zfp,n)
0106 plot(0,0,'+k')
0107 hold off
0108 axis([lbdrn ubdrn lbde ubde])
0109 caxis([zmin zmax])
0110 colormap jet
0111 colorbar
0112 xlabel('\delta_r_n (deg)')

```

```

0113 ylabel('\delta_e (deg)')
0114 title('C_y Polynomial Fit','Color','r')
0115 text(lbdrn-25,ubde+5,['Bias = ',num2str(bias),' removed'],'Color','b','FontSize',10)
0116
0117 subplot(2,2,4)
0118 surf(Xfp,Yfp,Zfp)
0119 caxis([zmin zmax])
0120 colormap jet
0121 colorbar
0122 axis([-15 25 -30 30 zmin zmax])
0123 xlabel('\delta_r_n (deg)')
0124 ylabel('\delta_e (deg)')
0125 zlabel('C_y')
0126 f = 3; %number of significant digits
0127 title(['C_y_w(\beta)+',...
0128     num2str(xp(1),f),'+',num2str(xp(2),f),'*\delta_e+',num2str(xp(3),f),...
0129     '*\delta_e^2+',num2str(xp(4),f),'*\delta_r_n+',num2str(xp(5),f),...
0130     '*\delta_r_n^2+',num2str(xp(6),f),'*\delta_e*\delta_r_n+',num2str(xp(7),f),...
0131     '*\delta_e^2*\delta_r_n+',num2str(xp(8),f),'*\delta_e*\delta_r_n^2+',...
0132     num2str(xp(9),f),'*\delta_e^2*\delta_r_n^2 (deg)
'],'Color','r','FontSize',6)
0133
0134 saveas(60,'Cy_polyfit_2d.jpg')
0135
0136 %Output from this code is as follows:

```

## Appendix C: Matlab Force and Moment Calculator

```

0001 function [L,D,Y,m_pitch,n_yaw,l_roll]=ForcesMoments(delta_e,delta_rn,V,alpha,beta,p,q,r)
0002 % Capt Travis Higgs
0003 % Thesis: Batcam Control Law Development
0004 % This function is used to calculate the forces and moments about the
0005 % BATCAM.
0006
0007 %Inputs:
0008 % delta_e    deg
0009 % delta_rn    deg
0010 % V          Velocity (ft/s)
0011 % alpha       Angle of Attack (rad)
0012 % beta       Angle of Sideslip (rad)
0013 % p          Roll Rate (rad/s)
0014
0015 %Outputs:
0016 % L          %Lift (lb)
0017 % D          %Drag (lb)
0018 % Y          %Side Force (lb)
0019 % m_pitch    %Pitching moment (ft-lb)
0020 % n_yaw      %Yawing moment (ft-lb)
0021 % l_roll     %Rolling moment (ft-lb)
0022
0023 global CLc CDc Cyc Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar u0 x0 Clb Cnb DF
0024
0025 %%%%%%%%%%%Calculate the Forces and Moments
0026 %%%%%%%%%%%
0027 % Create a vector of the 9 delta_e and delta_rn combinations as used in the
0028 % polynomial data fits...
0029 A1p = 1;
0030 A2p = delta_e;
0031 A3p = delta_e.^2;
0032 A4p = delta_rn;
0033 A5p = delta_rn.^2;
0034 A6p = delta_e.*delta_rn;
0035 A7p = delta_e.^2.*delta_rn;
0036 A8p = delta_e.*delta_rn.^2;
0037 A9p = delta_e.^2.*delta_rn.^2;
0038 Ap = [A1p; A2p; A3p; A4p; A5p; A6p; A7p; A8p; A9p]; %stack the elements vertically
0039 alphad = alpha*180/pi; %Rad to Deg for Force and Moment Calculations ONLY
0040 betad = beta*180/pi;
0041
0042 qbar = .5*rho*V^2; %Dynamic Pressure lbf/ft^2 or psf
0043
0044 eff=1; %Tail Effectiveness (as though the tail had more range of movement)
0045 %Determine coefficients based on delta_e, delta_rn, alpha, and beta and math models of
0046 %experimental data (all of these claculations based on degrees):
0047 CL = (-0.0002*alphad.^3-0.0014*alphad.^2+0.0924*alphad+0.7948)+eff*sum(CLc.*Ap); %Lift
0048 L = qbar*S*CL; %Lift (lb)
0049

```

```

0050 CD = (3E-05*alphanad.^3+0.0008*alphanad.^2+0.0052*alphanad+0.078)+eff*sum(CDc.*Ap); %Drag
Coeff
0051 D = qbar*S*CD; % Drag (lb)
0052
0053 %Cy = (0.0074*betad-0.0455)+sum(Cyc.*Ap); %Side Force Coeff (original)
0054 %Remove Cy bias of -.0455 from the beta fit to make it cross at 0,0.
0055 %Cy = (-0.0074*betad)+eff*sum(Cyc.*Ap); %Side Force Coeff
0056
0057 %Remove the 1st (const) term of the curve fit to allow zero coefficient at delta_rn=0
0058 Cy = (-0.0074*betad)+eff*sum(Cyc.*Ap);%-Cyc(1); %Side Force Coeff
0059 Y = qbar*S*Cy; %Side Force (lb)
0060
0061 % Cm has an offset taken out of it to put the values
0062 % realistically closer to zero (effectively shift the CG to match the data between
0063 % Cm_wing and the Cm changes due to tail deflections)
0064 Cm = (9E-06*alphanad.^4-7E-05*alphanad.^3-0.0018*alphanad.^2-
0.011*alphanad+0.0087)+eff*(sum(Cmc.*Ap)-Cmc(1));%Pitching Moment Coeff
0065 m_pitch = qbar*S*cbar*Cm; %Pitching moment (ft-lb)
0066
0067 %Cn = (0.0005*betad+0.0019)+sum(Cnc.*Ap);%Yawing Moment Coeff (Original)
0068 % Cn(beta) of the aircraft with zero tail deflections
0069 % has bias removed of +.0019 removed to make it cross at 0,0,
0070 %Cn = (0.0005*betad)+eff*sum(Cnc.*Ap);%Yawing Moment Coeff
0071
0072 % Cnb is the dihedral effect slope value passed in from the
0073 % C_l_polyfit_2d(or e).m function via the Trimmer.m where it's loaded and
0074 % made a global variable
0075 %Cn = (Cnb*betad)+eff*sum(Cnc.*Ap);%Yawing Moment Coeff
0076
0077 %Remove the 1st (const) term of the curve fit to allow zero coefficient at delta_rn=0
0078 Cn = (Cnb*betad)+eff*sum(Cnc.*Ap);%-Cnc(1);%Yawing Moment Coeff
0079 n_yaw = qbar*S*b*Cn; %Yawing moment (ft-lb)
0080
0081 %CLaw=.0825*pi/180; lamda=1; %This top chunk was used when mapping Clb as
0082 %a function of both alpha and beta before finding delta_e and delta_rn tail
0083 %effectiveness
0084 %Cl =
(xw(1)+xw(2)*alphanad+xw(3)*alphanad.^2+xw(4).*betad+xw(5)*betad.^2+xw(6)*alphanad.*betad+...
0085 % xw(7)*alphanad.^2.*betad+xw(8)*alphanad.*betad.^2+xw(9)*alphanad.^2.*betad.^2)+...
0086 % sum(Clc.*Ap)-(1/12*CLaw*(1+3*lamda)/(1+lamda))*p %Rolling Moment Coeff
0087 %Added roll damping term: Clp = -(1/12*CLaw*(1+3*lamda)/(1+lamda))*p;
0088
0089 %Disregard alpha effect on C_l
0090 DF=50; %Damping Factor for Rolling moment to account for flexible wing damping
0091 CLaw=(.0825*pi/180);
0092 lamda=1; %Elliptical wing is actually 0.45 that leads to 19% reduction
0093 %Clb
0094 Lbb=qbar*S*b/Ixx*Clb*betad; %sanity check for linmod beta coupling
0095 Cl = Clb*betad+eff*sum(Clc.*Ap)-DF*(1/12*CLaw*(1+3*lamda)/(1+lamda))*p; %Rolling Moment
Coeff
0096 l_roll = qbar*S*b*Cl; %Rolling moment (ft-lb)
0097
0098 return

```

## Appendix D: Matlab Trimmer for SLUF

```
0001 function x0=Trimmer_f()
0002 % Capt Travis Higgs
0003 % Thesis: Batcam Control Law Development
0004 % This function uses fmincon to optimize alpha, beta, thrust, delta_e, and
0005 % delta_rn for equilibrium: Steady Level Unaccelerated Flight (SLUF)
0006 % at a velocity of 30 ft/sec
0007 % Trimmer_f.m
0008
0009 % Version f has a modified cost function and graphical option for visual
0010 % analysis during optimization
0011
0012 % Dimensions      Units
0013 % mass            slugs
0014 % length          ft
0015 % area            ft^2
0016 % velocity        ft/s
0017 % acceleration    ft/s^2
0018 % density          slugs/ft^3
0019 % force           lbf
0020 % moments         lbf-ft
0021 % angles          calculations in radians
0022 % angular velocity rad/s
0023 % angular accel   rad/s^2
0024
0025 clear; clc; %clf;
0026 format short g
0027
0028 global CLc CDc Cyc Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar u0 x0 Clb Cnb C
0029
0030 % Get polynomial data fit coefficients (9 in each vector) from file...
0031 load CL_polyfit_2d.mat %Aircraft Lift Coefficient
0032 CLc = xp; clear xp
0033 load CD_polyfit_2d.mat %Aircraft Drag Coefficient
0034 CDc = xp; clear xp
0035 load Cy_polyfit_2d.mat %Aircraft Side Force Coefficient
0036 Cyc = xp; clear xp
0037 load Cm_polyfit_2d.mat %Aircraft Pitching Moment Coefficient
0038 Cmc = xp; clear xp
0039 load Cn_polyfit_2d.mat %Aircraft Yawing Moment Coefficient
0040 Cnc = xp; clear xp
0041 load C_l_polyfit_2e.mat %Aircraft Rolling Moment Coefficient
0042 Clc= xp; clear xp
0043
0044 %Define Constants
0045 rho = .002378;%slugs/ft^3 standard day density of air
0046 g = 32.17;%ft/s^2 gravitational constant
0047
0048 %Aircraft physical properties
0049 W = 0.79; %lb (360grams) weight of aircraft
0050 m = W/g; %slugs mass of aircraft
0051 Ixx = 8.96/(144*32.17);% slug-ft^2 roll mass moment of inertia
```



```

0052 Iyy = 30.72/(144*32.17);% slug-ft^2 pitch mass moment of inertia
0053 Izz = 32.52/(144*32.17);% slug-ft^2 yaw mass moment of inertia
0054 Ixz = 0;
0055 S = 0.65;%ft^2 planform area
0056 b = 2.0;%ft wing span
0057 cbar = 4.2/12;%ft wing mean geometric chord
0058
0059 %Define Initial Conditions:
0060 %STATE Equilibrim Values (SLUF)
0061 Vo = 30;% Velocity (ft/s) (same as 20.5 mph)
0062 gammao = 0*pi/180;% flight path angle (deg to rads)
0063 alphas = 0*pi/180;% angle of attack from (deg to rads)
0064 qo = 0;% Pitch Rate (rad/s)
0065 po = 0;% Roll Rate (rad/s)
0066 muo = 0;% Bank Angle (About Velocity Vector)
0067 betao = 0*pi/180;% angle of sideslip from (deg to rads)
0068 ro = 0;% Yaw Rate (rad/s)
0069 chio = 0*pi/180;% Heading angle (deg to rads)
0070 zetao = 0;% North Position (ft)
0071 etao = 0;% East Postion (ft)
0072 ho = 50;% Altitude (ft)
0073
0074 %INPUT Equilibrium Values (SLUF)
0075 To = 0.1;% Thrust
0076 delta_eo = 0;% Tail DEFLECTION (+) is down (deg)
0077 delta_rno = 0;% Tail ROTATION (+) is clockwise from aft view (deg)
0078
0079 %Create Initial Condition Vector
0080 s0(1) = Vo;
0081 s0(2) = gammao;
0082 s0(3) = alphas;
0083 s0(4) = qo;
0084 s0(5) = po;
0085 s0(6) = muo;
0086 s0(7) = betao;
0087 s0(8) = ro;
0088 s0(9) = chio;
0089 s0(10) = zetao;
0090 s0(11) = etao;
0091 s0(12) = ho;
0092
0093 s0(13) = To;
0094 s0(14) = delta_eo;
0095 s0(15) = delta_rno;
0096
0097 options=optimset('Display','iter','MaxIter',1000,'MaxFunEvals',5000); % Option to display output
0098
0099 %minimize J...
0100 s = fmincon(@myfun,s0,[],[],[],[],[],@mycon,options);
0101 %x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options,P1,P2, ...)
0102 %check J
0103 V = s(1)
0104 alpha = s(3);

```

```

0105 q    = s(4);
0106 p    = s(5);
0107 beta  = s(7);
0108 r     = s(8);
0109 T     = s(13)
0110 delta_e = s(14)
0111 delta_rn= s(15)
0112
0113 alphad = alpha*180/pi %Rad to Deg for Force and Moment Calculations ONLY
0114 betad  = beta*180/pi
0115
0116 [L,D,Y,m_pitch,n_yaw,l_roll] = ForcesMoments(delta_e,delta_rn,V,alpha,beta,p,q,r)
0117
0118 %Assign the initial states
0119 % x0(1) = 0;
0120 % x0(2) = 0;
0121 % x0(3) = 0;
0122
0123 x0(1) = s(1);
0124 x0(2) = s(2);
0125 x0(3) = s(3);
0126 x0(4) = s(4);
0127 x0(5) = s(5);
0128 x0(6) = s(6);
0129 x0(7) = s(7);
0130 x0(8) = s(8);
0131 x0(9) = s(9);
0132 x0(10) = s(10);
0133 x0(11) = s(11);
0134 x0(12) = s(12)
0135
0136
0137 %Assign the initial trim controls
0138 u0(1) = s(13);
0139 u0(2) = s(14);
0140 u0(3) = s(15)
0141
0142 return
0143
0144
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0145
0146 function J = myfun(s) % cost function
0147 global CLc CDc Cyclic Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar
0148 %clf;
0149 %Determine coefficients based on delta_e, delta_rn, alpha, and beta and math models of
0150 %experimental data:
0151 V    = s(1);
0152 alpha = s(3);
0153 q     = s(4);
0154 p     = s(5);
0155 beta  = s(7);

```

```

0156 r = s(8);
0157 T = s(13);
0158 delta_e = s(14);
0159 delta_rn= s(15);
0160
0161 [L,D,Y,m_pitch,n_yaw,l_roll] = ForcesMoments(delta_e,delta_rn,V,alpha,beta,p,q,r);
0162
0163 c1=(L+T*sin(alpha*pi/180)-W)^2; %L vs W (lbs)
0164 c2=(T*cos(alpha*pi/180)-D)^2; %T vs D (lbs)
0165 c3=abs(Y); %side force (lbs)
0166 c4=abs(m_pitch); %Pitching Moment (ft-lb)
0167 c5=abs(n_yaw); %Yaw Moment (ft-lb)
0168 c6=3*abs(l_roll); %Roll Moment (ft-lb)
0169 C=[c1 c2 c3 c4 c5 c6]; %Stack chunks into vector for plotting
0170 J=c1+c2+c3+c4+c5+c6; %Total Cost
0171
0172 %This bar graph shows how "hard" the optimizer is working, and what
0173 %portions of the cost are driving the optimization. It slows down the
0174 %computation, but is useful for development. Turn it off (comment out)
0175 %once you get your cost and constraint functions tweaked out.%
0176 % figure(1)
0177 % bar(C)
0178 % xlabel('L T Y m n l')
0179 % ylabel('cost')
0180 % pause(.02)
0181
0182 return
0183
0184
0185 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0186 function [c,ceq] = mycon(s) % constraints
0187 global CLc CDc Cyc Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar Cnb
0188 V = s(1);
0189 alpha = s(3);
0190 p = s(5);
0191 beta = s(7);
0192 T = s(13);
0193 delta_e = s(14);
0194 delta_rn= s(15);
0195
0196 alphad = alpha*180/pi; %Rad to Deg for Force and Moment Calculations ONLY
0197 betad = beta*180/pi;
0198
0199 % <= 0 inequality constraints
0200 c(1) = alphad-12; % -4<=alpha<=12 degrees
0201 c(2) = -alphad-4; %driven by C_l(alpha,beta)
0202 c(3) = betad-0; % -0<=beta<=0 degrees
0203 c(4) = -betad-0;
0204 c(5) = -T; % T=>0
0205 c(6) = delta_e-15; % -30<=delta_e<=15 degrees
0206 c(7) = -delta_e-30; %driven by data range

```

```

0207     c(8) = delta_rn-0;    % -0<=delta_rn<=0 degrees
0208     c(9) = -delta_rn-0;   % Data: -20<=delta_rn<=25 degrees
0209 %     c(10)= -V-30;       % 30<=V<=44 ft/s (OR 20.5<=V<=30 mph)
0210 %     c(11)= -V-44;
0211
0212 % = 0 equality constraints
0213     ceq(1) = V-30;        % True Airspeed, V=30 ft/sec
0214     ceq(2) = p;          % Roll Rate must be zero
0215 return
0216
0217
0218 % Resulting output from running the above code:

```

## Appendix E: Non-linear EOMs

```

0001 function xd=BATCAMsim_f(in)
0002 % Capt Travis Higgs
0003 % Thesis: Batcam Control Law Development
0004 % This function is called by a Simulink model (nonlinear_eom.mdl)
0005 % to solve the equations of motion for the BATCAM aircraft
0006 % BATCAMsim_f.m
0007
0008 %This code is adapted from Maj Paul Blue's Fortran code from 31 May 1994.
0009
0010 %Version d uses a different state variable order to better line up the
0011 %phugoid, short period, roll and dutch roll responses for linearization and
0012 %analysis.
0013
0014 % Dimensions      Units
0015 % mass            slugs
0016 % length          ft
0017 % area            ft^2
0018 % velocity        ft/s
0019 % acceleration    ft/s^2
0020 % density          slugs/ft^3
0021 % force           lbf
0022 % moments         lbf-ft
0023 % angles          calculations in radians (except polynomial data fits)
0024 % angular velocity rad/s
0025 % angular accel   rad/s^2
0026
0027 %clear; clc; %clf;
0028 format short g
0029
0030 global CLc CDc Cyc Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar u0 x0
0031
0032 %Define States
0033 % To      = u0(1);
0034 % delta_eo = u0(2);
0035 % delta_rno = u0(3);
0036 %
0037 % del_T      = in(1);      %Thrust
0038 % del_delta_e = in(2);      %Tail DEFLECTION (+) is down (deg)
0039 % del_delta_rn = in(3);      %Tail ROTATION (+) is clockwise from aft view (deg)
0040 %
0041 % T      = To+del_T;
0042 % delta_e = delta_eo+del_delta_e;
0043 % delta_rn= delta_rno+del_delta_rn;
0044
0045 T      = in(1);      %Thrust
0046 delta_e = in(2);      %Tail DEFLECTION (+) is down (deg)
0047 delta_rn = in(3);      %Tail ROTATION (+) is clockwise from aft view (deg)
0048
0049 V      = in(4);      % Velocity (ft/s)
0050 gamma   = in(5);      %flight path angle (rad)
0051 alpha   = in(6);      %angle of attack from (rad)

```

```

0052 q      = in(7);      %Pitch Rate (rad/s)
0053 p      = in(8);      %Roll Rate (rad/s)
0054 mu      = in(9);      %Bank Angle (About Velocity Vector) (rad)
0055 beta    = in(10);     %angle of sideslip (rad)
0056 r      = in(11);     %Yaw Rate (rad/s)
0057 chi     = in(12);     %Heading angle (rads)
0058 zeta    = in(13);     %North Position (ft)
0059 eta     = in(14);     %East Postion (ft)
0060 h       = in(15);     %Altitude (ft)
0061
0062 [L,D,Y,m_pitch,n_yaw,l_roll] = ForcesMoments(delta_e,delta_rn,V,alpha,beta,p,q,r);
0063
0064 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Non-linear Equations Of Motion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0065 %Currently assumes Ixz = 0 and no moments due to propeller...
0066 V_dot    = 1/m*(-D*cos(beta)+Y*sin(beta)+T*cos(beta)*cos(alpha))-g*sin(gamma);% Vector
Accel (ft/s^2)
0067 gamma_dot = 1/(m*V)*(-D*sin(beta)*sin(mu)-
Y*sin(mu)*cos(beta)+L*cos(mu)+T*(cos(mu)*sin(alpha)+sin(mu)*sin(beta)*cos(alpha)))-
g/V*cos(gamma); % Flight Path Angle Rate (rad/s)
0068
0069 alpha_dot = q-tan(beta)*(p*cos(alpha)+r*sin(alpha))-
1/(m*V*cos(beta))*(L+T*sin(alpha))+g*cos(gamma)*cos(mu)/(V*cos(beta));% Alpha Rate (rad/s)
0070 q_dot    = m_pitch/Iyy+1/Iyy*(Izz*p*r-Ixx*r*p); % Pitch Accel (rad/s^2)
0071
0072 p_dot    = l_roll/Ixx+1/Ixx*(Iyy*r*q-Izz*q*r); % Roll Accel (rad/s^2)
0073 mu_dot   = 1/cos(beta)*(p*cos(alpha)+r*sin(alpha))+1/(m*V)*(D*sin(beta)*cos(mu)*...
0074 tan(gamma)+Y*tan(gamma)*cos(mu)*cos(beta)+L*(tan(beta)+tan(gamma)*sin(mu))+...
0075 T*(sin(alpha)*tan(gamma)*sin(mu)+sin(alpha)*tan(beta)-cos(alpha)*tan(gamma)*...
0076 cos(mu)*sin(beta))-g/V*cos(gamma)*cos(mu)*tan(beta); % Bank Angle (about vel vector)
Rate (rad/s)
0077
0078 beta_dot = -r*cos(alpha)+p*sin(alpha)+1/(m*V)*(D*sin(beta)+Y*cos(beta)-
T*sin(beta)*cos(alpha))+g/V*cos(gamma)*sin(mu);% Beta Rate (rad/s)
0079 r_dot    = n_yaw/Izz+1/Izz*(Ixx*p*q-Iyy*p*q); % Yaw Accel (rad/s^2)
0080
0081 chi_dot  =
1/(m*V*cos(gamma))*(D*sin(beta)*cos(mu)+Y*cos(mu)*cos(beta)+L*sin(mu)+T*(sin(mu)*sin(alpha)-
cos(mu)*sin(beta)*cos(alpha))); % Heading Angle Rate (rad/s)
0082 zeta_dot = V*cos(gamma)*cos(chi); % North Potition Rate (ft/s)
0083 eta_dot  = V*cos(gamma)*sin(chi); % East Position Rate (ft/s)
0084 h_dot    = V*sin(gamma); % Altitude Rate (ft/s)
0085
0086 % Assign values to the stacked state derivative vector:
0087 xd(1) = V_dot; % Vector Accel (ft/s^2)
0088 xd(2) = gamma_dot; %Flight Path Angle Rate (rad/s)
0089
0090 xd(3) = alpha_dot; % Alpha Rate (rad/s)
0091 xd(4) = q_dot; %Pitch Accel (rad/s^2)
0092
0093 xd(5) = p_dot; %Roll Accel (rad/s^2)
0094 xd(6) = mu_dot; %Bank Angle (about vel vector) Rate (rad/s)
0095

```

```

0096 xd(7) = beta_dot; %Beta Rate (rad/s)
0097 xd(8) = r_dot; %Yaw Accel (rad/s^2)
0098
0099 xd(9) = chi_dot; %Heading Angle Rate (rad/s)
0100 xd(10) = zeta_dot; %North Potition Rate (ft/s)
0101 xd(11) = eta_dot; %East Position Rate (ft/s)
0102 xd(12) = h_dot; %Altitude Rate (ft/s)
0103
0104
0105 return

```

## Appendix F: Dynamic Output (Flight Path & Strip Charts)

```

0001 %function PlotOutput_d()
0002 % Capt Travis Higgs
0003 % Thesis: Batcam Control Law Development
0004 % This code uses the output from the BATCAM Simulink model to plot results
0005 % for visual evaluation. It saves the figure(s) to .jpg file(s) for
0006 % archiving and later analysis.
0007 % PlotOutput_d.m
0008
0009 % Version d of PloutOutput uses a different order of state variables that
0010 % categorizes
0011
0012 % Dimensions      Units
0013 % mass            slugs
0014 % length          ft
0015 % area            ft^2
0016 % velocity        ft/s
0017 % acceleration    ft/s^2
0018 % density          slugs/ft^3
0019 % force            lbf
0020 % moments          lbf-ft
0021 % angles           deg   Few people THINK in rads!
0022 % angular velocity deg/s
0023 % angular accel    deg/s^2
0024
0025 clf; %clear; clc;
0026 format short g
0027
0028 global CLc CDc Cyclic Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar u0 x0 Clb Cnb DF
0029
0030 %%%%%%%%%%% Get and calculate all necessary values
0031 %%%%%%%%%%%
0031 load SimOutput.mat      % States for corresponding times
0032
0033 t      = ans(1,:); %Pull Time Vector
0034
0035 %Define States
0036 V      = ans(2,:); % Velocity (ft/s)
0037 gamma  = ans(3,:)*180/pi; %flight path angle (deg)
0038 alpha  = ans(4,:)*180/pi; %angle of attack (deg)
0039 q      = ans(5,:)*180/pi; %Pitch Rate (deg/sec)
0040 p      = ans(6,:)*180/pi; %Roll Rate (deg/sec)
0041 mu     = ans(7,:)*180/pi; %Bank Angle (About Velocity Vector) (deg)
0042 beta   = ans(8,:)*180/pi; %angle of sideslip (deg)
0043 r      = ans(9,:)*180/pi; %Yaw Rate (deg/sec)
0044 chi    = ans(10,:)*180/pi; %Heading angle (deg)
0045 zeta   = ans(11,:); %North Position (ft)
0046 eta    = ans(12,:); %East Postion (ft)
0047 h      = ans(13,:); %Altitude (ft)
0048
0049 To     = u0(1); %Thrust
0050 delta_eo = u0(2); %Tail DEFLECTION (+) is down (deg)

```



```

0051 delta_rno = u0(3);    %Tail ROTATION (+) is clockwise from aft view (deg)
0052
0053
0054 %%%%%%%%%%%%%%% Flight Path
%%%%%%%%%%%%%%
0055 figure(90)
0056 clf;
0057 subplot(2,2,1)    %Top: God's eye view
0058 plot(eta,zeta,'-r') %solid red line is Track from starting point (no wind)
0059 N = 1.1*max(zeta);    %Max north distance (ft)
0060 E = 1.1*max(abs(eta));    %Max east distance (ft)
0061 axis([-E E 0 N])
0062 axis square
0063 xlabel('W    (ft)    E');
0064 ylabel('N');
0065 set(get(gca,'YLabel'),'Rotation',0.0) %rotates the text of the y-axis label
0066 title('Top View')
0067 grid on
0068
0069 %These are all purposely displayed in the upper right quadrant of the figure
0070 f = 3; %number of significant digits
0071 text (E,1.1*N,'Initial Conditions:')
0072 text (1.2*E,1.0*N,'SLUF') %This could be variable with if else statements
0073 text (1.2*E,.9*N,['V = ',num2str(V(1),f),' ft/s'])
0074 text (1.2*E,.8*N,['\beta = ',num2str(beta(1),f),' deg'])
0075 text (1.2*E,.7*N,['\alpha = ',num2str(alpha(1),f),' deg'])
0076 text (1.2*E,.6*N,['\mu = ',num2str(mu(1),f),' deg'])
0077 text (1.2*E,.5*N,['\gamma = ',num2str(gamma(1),f),' deg'])
0078 text (1.2*E,.4*N,['\chi = ',num2str(chi(1),f),' deg'])
0079
0080 text (1.2*E,.25*N,['T = ',num2str(To,f),' lbs'])
0081 text (1.2*E,.15*N,['\delta_e = ',num2str(delta_eo,f),' deg'])
0082 text (1.2*E,.05*N,['\delta_r_n = ',num2str(delta_rno,f),' deg'])
0083
0084 text (3*E,1.0*N,['Cl_\beta= ',num2str(Clb,4)])
0085 text (3*E,0.8*N,['Cn_\beta= ',num2str(Cnb,4)])
0086 text (3*E,0.6*N,['DF = ',num2str(DF,2)])
0087
0088 subplot(2,2,3)    %Aft View
0089 plot(eta,h,'+r')    %red cross Altitude and Lateral Deviation
0090 Alt = 2*max(h);    %Max altitude (ft)
0091 axis([-E E 0 Alt])
0092 axis square
0093 xlabel('W    (ft)    E');
0094 ylabel('Alt');
0095 set(get(gca,'YLabel'),'Rotation',0.0) %rotates the text of the y-axis label
0096 title('Aft View')
0097 grid on
0098
0099 subplot(2,2,4)    %Side View
0100 hold on
0101 plot(t,h,'-r')    %solid red line is Track from starting point (no wind)
0102 plot(t,V,'-b')    %dashed blue line is True Airspeed, V, (ft/s)

```

```

0103 Alt = 2*max(h);      %Max altitude (ft)
0104 tf = max(t)          %Max time (sec)
0105 axis([0 tf 0 Alt])
0106 xlabel('time (sec)');
0107 legend('Altitude (ft)', 'True Airspeed (ft/s)')
0108 title('Side View')
0109 grid on
0110
0111 l = Clb*100;
0112 n = Cnb*100;
0113 saveas(gcf,['OL_SimFltPth_Clb',num2str(l),'_Cnb',num2str(n),'_DF',num2str(DF),'.jpg'])
0114
0115
0116 figure(91)
0117 clf;
0118 plot3(eta,zeta,h)
0119 xlabel('W (ft) E')
0120 ylabel('N')
0121 zlabel('altitude')
0122 title('3D plot of aircraft spatial position (ft)')
0123 title(['3D plot of aircraft spatial position (ft)  CL_\beta= ',num2str(Clb,4),' Cn_\beta= ',num2str(Cnb,4),' DF= ',num2str(DF,2)])
0124 grid on
0125 saveas(gcf,['OL_3D_SimFltPth_Clb',num2str(l),'_Cnb',num2str(n),'_DF',num2str(DF),'.jpg'])
0126
0127 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Strip Charts
0128 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0129 figure(95)
0130 clf;
0131 subplot(12,1,1)
0132 plot(t,V)
0133 %title('Rates: deg/s Angles: deg Distance: ft Velocity: ft/s')
0134 title(['Rates: deg/s Angles: deg Distance: ft Velocity: ft/s CL_\beta= ',num2str(Clb,4),' Cn_\beta= ',num2str(Cnb,4),' DF= ',num2str(DF,2)])
0135 ylabel('V');
0136 axis([0 tf 0 50])
0137 grid on
0138
0139
0140 subplot(12,1,2)
0141 plot(t,gamma)
0142 ylabel('\gamma');
0143 axis([0 tf -10 10])
0144 grid on
0145
0146 subplot(12,1,3)
0147 plot(t,q)
0148 plot(t,alpha)
0149 ylabel('\alpha');
0150 axis([0 tf -5 15])
0151 grid on
0152

```

```

0153 subplot(12,1,4)
0154 plot(t,q)
0155 ylabel('q');
0156 axis([0 tf -45 45])
0157 grid on
0158
0159 subplot(12,1,5)
0160 plot(t,p)
0161 ylabel('p');
0162 axis([0 tf -360 360])
0163 grid on
0164
0165 subplot(12,1,6)
0166 plot(t,mu)
0167 ylabel('\mu');
0168 axis([0 tf -90 90])
0169 grid on
0170
0171 subplot(12,1,7)
0172 plot(t,beta)
0173 ylabel('\beta');
0174 axis([0 tf -45 45])
0175 grid on
0176
0177 subplot(12,1,8)
0178 plot(t,r)
0179 ylabel('r');
0180 axis([0 tf -90 90])
0181 grid on
0182
0183 subplot(12,1,9)
0184 plot(t,chi)
0185 ylabel('\chi');
0186 axis([0 tf -360 360])
0187 grid on
0188
0189 subplot(12,1,10)
0190 plot(t,zeta)
0191 ylabel('N');
0192 axis([0 tf 0 N])
0193 grid on
0194
0195 subplot(12,1,11)
0196 plot(t,eta)
0197 ylabel('E');
0198 grid on
0199
0200 subplot(12,1,12)
0201 plot(t,h)
0202 xlabel('time (sec)');
0203 ylabel('h');
0204 axis([0 tf 0 75])
0205 grid on

```

```

0206
0207 saveas(gcf,['OL_Strips_Clb',num2str(l),'_Cnb',num2str(n),'_DF',num2str(DF),'.jpg'])
0208 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problem Children
0209 figure(96)
0210 clf;
0211 subplot(4,1,1)
0212 plot(t,p)
0213 %title('Problem Children (Rates: deg/s Angles: deg Distance: ft Velocity: ft/s)')
0214 title(['Crucial States Cl_\beta= ',num2str(Clb,4),' Cn_\beta= ',num2str(Cnb,4),' DF= ',num2str(DF,2)])
0215 ylabel('Roll Rate, p (deg/s)');
0216 axis([0 tf -10 10])
0217 grid on
0218
0219 subplot(4,1,2)
0220 plot(t,mu)
0221 ylabel('Bank Angle, \mu (deg)');
0222 axis([0 tf -90 90])
0223 grid on
0224
0225 subplot(4,1,3)
0226 plot(t,beta)
0227 ylabel('Sideslip, \beta (deg)');
0228 axis([0 tf -45 45])
0229 grid on
0230
0231 subplot(4,1,4)
0232 plot(t,r)
0233 xlabel('time (sec)');
0234 ylabel('Yaw Rate, r (deg/s)');
0235 axis([0 tf -90 90])
0236 grid on
0237
0238 saveas(gcf,['OL_CrucialStates_Clb',num2str(l),'_Cnb',num2str(n),'_DF',num2str(DF),'.jpg'])
0239

```

## Appendix G: Matlab Linearization and Eigenvalues

```

0001 %function BATCAM_lin()
0002 % Capt Travis Higgs
0003 % Thesis: Batcam Control Law Development
0004 % This code linearizes the BATCAM nonlinear EOM simulink model using
0005 % the linmod command. It then checks the A matrix eigenvalues for stability
0006 % BATCAM_lin.m
0007
0008 % Dimensions      Units
0009 % mass            slugs
0010 % length          ft
0011 % area            ft^2
0012 % velocity        ft/s
0013 % acceleration    ft/s^2
0014 % density          slugs/ft^3
0015 % force           lbf
0016 % moments         lbf-ft
0017 % angles          deg   Few people THINK in rads!
0018 % angular velocity deg/s
0019 % angular accel    deg/s^2
0020
0021 clear; clc; %clf;
0022 format short g
0023
0024 global CLc CDc Cyc Cmc Cnc Clc xw rho g W m Ixx Iyy Izz Ixz S b cbar u0 x0
0025
0026 utest=u0
0027 xtest=x0
0028 [A,B,C,D] = linmod('BATCAM_nonlin_g');% ,xtest,utest)
0029 % [A2,B2,C2,D2] = linmod('BATCAM_nonlin_f_test');% ,x0,u0) Verification that
0030 % they are equivalent
0031 % [A,B,C,D] = linmod('sys', x, u) obtains the linearized model of sys
0032 % around an operating point with the specified state variables
0033 % x and the input u. If you omit x and u, the default values are zero.
0034
0035 % A(5,7)=-1.2946; Replace the large coupling value of -74
0036 % discovered it is the result of linmod perturbing in RADIANS about the
0037 % trim condition. The force and moment coefficients are determined using
0038 % DEGREES, but the resulting coefficients are non-dimensional
0039
0040 % Pull the sub-matrices from the big A matrix...
0041 Aph = A(1:2,1:2) %phugoid
0042 Asp = A(3:4,3:4) %short period
0043 Aroll = A(5:6,5:6) %roll
0044 Aadr = A(7:8,7:8) %dutch roll
0045 Alat_dir= A(5:8,5:8) %coupling
0046
0047 % Calculate the eigenvalues of those sub matrices
0048 eig_ph = eig(Aph)
0049 eig_sp = eig(Asp)
0050 eig_roll = eig(Aroll)
0051 eig_dr = eig(Aadr)

```

```

0052 eig_lat_dir = eig(Alat_dir)
0053 eig_A = eig(A)
0054 worst_eig_A = max(real(eig(A)))
0055
0056 save BATCAM_linmod A B C D Aph Asp Aroll Adr -ascii -tabs; %save for manipulation in Excel
0057
0058 %%%%%%%%%%% Determine a Linear Controller
%%%%%%%%%%
0059 % No success with these--quick attempts out of curiosity
0060 %rank(B)
0061 % p=[-.75,-.75,-1,-1,-1,-2,-2,-2,-3,-3,-3];
0062 % K=place(A,B,p)
0063 % % Aplace=A-B*K
0064 % % eig(Aplace)
0065 % %
0066 % %%%%%%%%%%% LQR Method %%%%%%%%%%%
0067 % Q=100*eye(12);
0068 % %R=[.01 0 0;0 100 0;0 0 100];
0069 % R=eye(3);
0070 %
0071 % Klqr=lqr(A,B,Q,R)
0072 % Alqr=A-B*Klqr
0073 % eig(Alqr)
0074 %
0075 % x_disturbance = .1*rand(12,1)
0076 %
0077 % %plat_dir=[p1,p2,p3,p4]
0078 % %Kph=1
0079
0080 return

```

## Appendix H: Formatted State Space System

These A,B,C,D matrices represent the linearized version of the BATCAMsim\_g.m and BATCAM\_nonlin\_g.mdl  
for  $\delta m=0^\circ$  Bias Removal & Mirroring,  $Cn\beta = +.0018$ , and  $DF=50$

A Matrix													B Matrix												
ph	V	-0.2351	-32.17	-14.904	0	0	0	-0.000632	0	0	0	0	V	40.6922	0.00465	0.00157	δT								
	γ	0.07103	0	4.63326	0	0	2.108E-05	0	0	0	0	0	γ	0.05147	0.00531	-0.0002		δe							
sp	α	-0.071	0	-4.6333	1	0	0	0	0	0	0	0	α	-0.0515	-0.0053	0.00017	δm								
	q	-0.0022	0	-40.9	0	0	0	0	0	0	0	0	q	0	-0.3994	0.01007									
roll	p	-3E-18	0	0	0	-8.6302	0	-74.17685	0	0	0	0	p	0	3.5E-18	-0.0091									
	μ	0	-2.11E-05	2.1E-34	0	0.99928	0	-0.004791	0.0379166	0	0	0	μ	0	0	0									
dr	β	-7E-07	0	0	0	0.03792	1.0723333	-0.399961	-0.999281	0	0	0	β	0	9.3E-06	-9E-05									
	r	0.00081	0	0	0	0	0	20.437411	0	0	0	0	r	0	-0.0055	0.00343									
χ		-7E-07	0	0	0	0	1.0675418	-0.399961	0	0	0	0	χ	0	9.3E-06	-9E-05									
	ζ	1	0	0	0	0	0	0	0	0	0	0	ζ	0	0	0									
	η	1.9E-32	0	0	0	0	0	0	0	0	30	0	η	0	0	0									
	h	0	30	0	0	0	0	0	0	0	0	0	h	0	0	0									

Eigenvalues: -0.11756 ± 1.5071i, -2.3166 ± 5.961i, 0, -8.6302, -0.19998 ± 4.5147i

-9.163  
0.052394 ± 5.2824i  
0.028046

C Matrix

1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1

D Matrix

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

## Appendix I: Modified Data for $C_I$ , $C_n$ , and $C_Y$ (Zeroed and Mirrored)

This file used to make data2d.prn that is used by C_I_ployfit_2e.m										
Negative delta_rn values discarded and the positive delta_rn values mirrored about zero and the bias of +.00837 removed										
$\beta$	$\delta_e$	$\delta_m$	Tail	abalize	Uoo	alpha_c	$C_I$	$C_I_{unb}$	$C_n$	$C_{Yr}$
0	-30	-25	1	1	29.7774	4.835632	-0.012186964	-0.003818255	0.025297389	-0.105732491
0	-20	-25	1	1	29.8224	4.849883	-0.010084244	-0.001715535	0.014498408	-0.078460878
0	-10	-25	1	1	29.7123	4.866835	-0.008866263	-0.000497554	0.009714141	-0.066257496
0	0	-25	1	1	29.7370	4.897578	-0.008304904	6.3805E-05	0.000998308	-0.043363796
0	14	-25	1	1	29.7877	4.928339	-0.007047416	0.001321293	-0.009602023	-0.015745275
0	-30	-14	1	1	29.7853	4.832392	-0.010226352	-0.001857643	0.017371209	-0.084631176
0	-20	-14	1	1	29.7561	4.850262	-0.009266067	-0.000897358	0.010051454	-0.066241049
0	-10	-14	1	1	29.7188	4.866919	-0.008414259	-4.55498E-05	0.007508635	-0.060108838
0	0	-14	1	1	29.7536	4.898680	-0.00820964	0.000159069	0.001150916	-0.043460087
0	14	-14	1	1	29.7692	4.932976	-0.007488615	0.000880094	-0.00679415	-0.023293459
0	-30	-7	1	1	29.8009	4.828155	-0.008587115	-0.000218406	0.008754032	-0.061611454
0	-20	-7	1	1	29.7359	4.850439	-0.008345035	2.36746E-05	0.004569762	-0.051336499
0	-10	-7	1	1	29.7114	4.867302	-0.007862984	0.000505725	0.004300043	-0.051167973
0	0	-7	1	1	29.7517	4.899913	-0.008256375	0.000112334	0.001404513	-0.043759759
0	14	-7	1	1	29.8023	4.934323	-0.007991971	0.000376739	-0.003525992	-0.032239343
0	-30	0	1	1	29.8113	4.827601	0.008370000	0	0.002027854	-0.04531833
0	-20	0	1	1	29.7287	4.850308	0.008370000	0	0.002027854	-0.04531833
0	-10	0	1	1	29.7028	4.867591	0.008370000	0	0.002027854	-0.04531833
0	0	0	1	1	29.7564	4.901187	0.008368709	0	0.002027854	-0.04531833
0	14	0	1	1	29.8450	4.934756	0.008368709	0	0.002027854	-0.04531833
0	-30	7	1	1	29.8009	4.828155	0.008587115	0.000218406	0.008754032	-0.061611454
0	-20	7	1	1	29.7359	4.850439	0.008345035	-2.36746E-05	0.004569762	-0.051336499
0	-10	7	1	1	29.7114	4.867302	0.007862984	-0.000505725	0.004300043	-0.051167973
0	0	7	1	1	29.7517	4.899913	0.008256375	-0.000112334	0.001404513	-0.043759759
0	14	7	1	1	29.8023	4.934323	0.007991971	-0.000376739	-0.003525992	-0.032239343
0	-30	14	1	1	29.7853	4.832392	0.010226352	0.001857643	0.017371209	-0.084631176
0	-20	14	1	1	29.7561	4.850262	0.009266067	0.000897358	0.010051454	-0.066241049
0	-10	14	1	1	29.7188	4.866919	0.008414259	4.55498E-05	0.007508635	-0.060108838
0	0	14	1	1	29.7536	4.898680	0.00820964	-0.000159069	0.001150916	-0.043460087
0	14	14	1	1	29.7692	4.932976	0.007488615	-0.000880094	-0.00679415	-0.023293459
0	-30	25	1	1	29.7774	4.835632	0.012186964	0.003818255	0.025297389	-0.105732491
0	-20	25	1	1	29.8224	4.849883	0.010084244	0.001715535	0.014498408	-0.078460878
0	-10	25	1	1	29.7123	4.866835	0.008866263	0.000497554	0.009714141	-0.066257496
0	0	25	1	1	29.7370	4.897578	0.008304904	-6.3805E-05	0.000998308	-0.043363796
0	14	25	1	1	29.7877	4.928339	0.007047416	-0.001321293	-0.009602023	-0.015745275



## Bibliography

- Ashby, D.L. "Potential Flow Theory and Operation Guide for the Panel Code PMARC\_14," NASA TM-1999-209582, December 1999.
- Bryson, Arther E. Jr. *Dynamic Optimization*. Menlo Park CA. Addison Wesley Longman, Inc., 1999.
- DeLuca, Anthony M. *Experimental Investigation into the Aerodynamic Performance of both Rigid and Flexible Wing Structured Micro-Air-Vehicles*. MS thesis, AFIT/GAE/ENY/04-M06. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 2004 (ADA424137).
- Hoey, Robert G., *Research on the Stability and Control of Soaring Birds*. AIAA Paper 1992-4122. Lancaster CA, 1992.
- Honeywell Technology Center. "Application of Multivariable Control Theory to Aircraft Control Laws: Final Report, March 1993-March 1996". Contract F33615-92-C-3607. Paper WL-TR-96-3099. Minneapolis MN May 1996 (ADA315259).
- Kiemele, Mark J., Stephen R. Schmidt, and Ronald J. Berdine. *Basic Statistics: Tools for Continuous Improvement* (Fourth Edition). Colorado Springs CO: Air Academy Press & Associates, 1997.
- Leveron, Troy A. *Characterization of a Rotary Flat Tail as a Spoiler and Parametric Analysis of Improving Directional Stability in a Portable UAV*. MS thesis, AFIT/GAE/ENY/05-J06. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, June 2005.
- Matlab*. Version 6.5.0 Release 13, IBM, disk. Computer software. MathWorks, Inc, Natick MA, 2002
- Nelson, Robert C. *Flight Stability And Automatic Control* (Second Edition). Madison WI: McGraw-Hill, 1998.
- Parga, Jose R. Rivera. *Wind Tunnel Investigation of the Static Stability and Control Effectiveness of a Rotary Tail in a Portable UAV*. MS thesis, AFIT/GAE/ENY/04-D02. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, December 2004 (ADA430131).

- Raymer, Daniel P. *Aircraft Design: A conceptual Approach*. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1992.
- Roskam, Jan. *Airplane Flight Dynamics And Automatic Flight Controls: Part I*. Lawrence KS: Design, Analysis and Research Corporation, 1995.
- Simulink*. Version 5.0 Release 13, IBM, disk. Computer software. MathWorks, Inc, Natick MA, 2002
- Stevens, Brian L. and Frank L. Lewis. *Aircraft Control and Simulation* (Second Edition). Hoboken NJ: John Wiley & Sons, Inc, 2003.
- Waszak, Martin R., Luther N. Jenkins, and Peter Ifju. "Stability and Control Properties of an Aeroelastic Fixed Wing Micro Aerial Vehicle," *Proceedings of the IAAA Atmospheric Flight Mechanics Conference*. 2001-4005. Montreal Canada: American Institute of Aeronautics and Astronautics, Inc., 2001.
- Waszak, Martin R., John B. Davidson, and Peter Ifju. "Simulation and Flight Control of an Aeroelastic Fixed Wing Micro Aerial Vehicle," *Proceedings of the IAAA Atmospheric Flight Mechanics Conference*. 2002-4875. Monterey CA: American Institute of Aeronautics and Astronautics, Inc., 2002.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) 22-12-2005		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Dec 2004 - Dec 2005	
4. TITLE AND SUBTITLE  Modeling, Stability, and Control of a Rotatable Tail on a Micro Air Vehicle				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)  Higgs, Travis J., Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/GAE/ENY/06-05	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Lab / MNAV Attn: Lt. Michael Simonich (for Maj. John Anttonen ) 1865 4th Street Eglin AFB, FL 32542 DSN: 872-8876 ext. 3335				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/MNAV	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT  APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This research uses existing experimental wind tunnel data to develop a non-linear model that is used to characterize the stability of a flexible wing Micro Air Vehicle (MAV) with a rotatable tail. The experimental data are curve fit based on either angle of attack or angle of sideslip, and the coupled effect of tail rotation and tail deflection on the force and moment coefficients. Static optimization trims the input and state variables for Steady Level Unaccelerated Flight (SLUF). The resulting initial conditions are fed to an open loop non-linear Simulink/Matlab simulation. The study found that the bare MAV design is unstable, but parametric studies identified practical modifications that could be made to the MAV to improve its open loop stability characteristics. The study found that the coupling affect due to the dihedral derivative, $Cl_\beta$ , played a large role in destabilizing the lateral-directional dynamics and a feedback Stability Augmentation System is required for flight.					
15. SUBJECT TERMS MAV, Rotatable Tail, Curve Fit, Non-linear, Simulation, Optimization, Lateral-Directional, Dihedral					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Mark F. Reeder
U	U	U	UU	132	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4530 (mark.reeder@afit.edu)